DOCUMENT RESUME

ED 346 817 IR 014 351

TITLE	An Introduction to Computer Science (Secondary Grades). Volume 2.
INSTITUTION	New York City Board of Education, Brooklyn, N.Y.
REPORT NO	ISBN-88315-570-2
PUB DATE	85
NOTE	319p.; For volume 1, see IR 014 350.
AVAILABLE FROM	New York City Board of Education, Curriculum Unit, Room 310, P.S. 206, Neck Road and East 22nd Street, Brooklyn, NY 11229.
PUB TYPE	Guides - Classroom Use - Teaching Guides (For Teacher) (052)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Literacy; *Computer Science Education; Curriculum Guides; Glossaries; High Schools; *Microcomputers; *Programing; Programing Languages; Special Education; Spreadsheets; Telecommunications

ABSTRACT

The second volume of a two-volume computer science curriculum guide, reflecting the technology of the 1980s, presents 66 lessons on advanced computer literacy topics for high school students. Topics include advanced programing concepts in BASIC, spreadsheet management, telecommunications, writing a computer-assisted-instruction program, computers in society, other number systems, programing without BASIC (an introduction to compilers, machine language, assembly language, higher level languages, LOGO, and Pascal), and some useful mathematical topics. Performance objectives, vocabulary, motivation, aim, development, application, summary, and exercises are included for each lesson. The appendixes include: (1) a discussion of computers and special education students; (2) a list of 50 books for further reading; (3) a glossary of computer terms; (4) a directory of educational software vendors; (5) a list of 23 computing magazines; (6) a directory of popular hardware vendors; and (7) a brief description of primary sources for grant information. (MES)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED346817

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
This document has been reproduced as received from the person or organization originating it.
Minor changes have been made to improve reproduction quality.
Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

Computer Science

IR014 351

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
Dorothy Klein
TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

VOLUME 2

n of Curriculum and Instruction

BOARD OF EDUCATION CITY OF NEW YORK

Office of Curriculum Development and Support

BEST COPY AVAILABLE

AN INTRODUCTION TO Computer Science (SECONDARY GRADES)

VOLUME 2

EXPERIMENTAL EDITION

New York City Board of Education

James F. Regan, *President*
Miguel O. Martinez, *Vice President*

Amelia Ashe
Joseph G. Barkan
Stephen R. Franse
Irene Impellizzeri
Marjorie A. Lewis

Nathan Quinones, *Chancellor*

Charles I. Schonhaut, *Deputy Chancellor*

Jerald Posman, *Deputy Chancellor for Financial Affairs*

Louise Latty, *Chief Executive for Instruction*

Charlotte Frank, *Executive Director*
Division of Curriculum and Instruction

AN INTRODUCTION TO Computer Science (SECONDARY GRADES)

VOLUME 2

NEW YORK CITY BOARD OF EDUCATION

Division of Curriculum and Instruction

Office of Curriculum Development and Support

EXPERIMENTAL EDITION

Copyright ©1985
by the Board of Education of the City of New York

Application for permission to reprint any section of this material should be made to the Chancellor, 110 Livingston Street, Brooklyn, N.Y. 11201. Reprint of any section of this material shall carry the line, "Reprinted from *An Introduction to Computer Science (Secondary Grades), Volume 2* by permission of the Board of Education of the City of New York."

New York City Public Schools should order additional copies of this publication from the Bureau of Supplies.

ISBN No. 88315-570-2

Curric. No. 00-6090-82

Other persons and institutions may obtain copies of this publication from the Curriculum Production Unit of the Office of Curriculum Development and Support, Room 613, 131 Livingston Street, Brooklyn, N.Y. 11201. For information call (718) 596-4902.

FOREWORD

In 1983 the Division of Curriculum and Instruction published computer literacy curriculum guides for both elementary and secondary schools. That project was the first step in a comprehensive plan for computer education in New York City. Computer technology is new and is changing daily. We as educators are faced with a common challenge: how to use this exciting and powerful technology in a way that will meet the changing needs of the students and will continue to accommodate changes as they occur. The challenge is a great one, but the potential for these tools is even greater.

We know that computers can ignite the imagination and extend the limits of human learning. They enable their users to view the world in new ways. When students interact with computers they take charge of their own learning and learn more through this interaction. The problem of time on task disappears as students work with computers to solve problems in various ways. In short, computers are tools used by humans for human endeavors.

This two-volume computer science curriculum guide is the next step in the New York City plan for computer education. It has been designed as a support document for teachers whose students have already completed earlier computer courses in the sequence. We hope that this guide will aid teachers and students in keeping pace with the challenges of our information age.

Charlotte Frank

Charlotte Frank
Executive Director
Division of Curriculum and Instruction

ACKNOWLEDGMENTS

An Introduction to Computer Science (Secondary Grades), Volume 2 is a project of the Division of Curriculum and Instruction, Charlotte Frank, Executive Director.

Morris Freedman, Director of the Office of Curriculum Development and Support, provided overall supervision of this project.

Michael Ryan, Director of the Computer and Information Sciences Unit provided supervision and overall coordination. Joseph Eaione, Assistant Director of the Computer and Information Sciences Unit, gave impetus and guidance to this project by supervising the writing teams, arranging for the planning and production meetings, and screening the materials that were produced by the writers.

Stephen Radin of the Computer and Information Sciences Unit is the principal author and was responsible for the editing and sequencing of this guide.

The dedicated professionals listed below were involved in the writing and review of this document.

Nicholas Androu - Elizabeth Blackwell JHS 210, Ozone Park, New York
Laney Dahl - Curtis High School, Staten Island, New York
Elaine Fitzpatrick - William A. Morris IS 61, Staten Island
Larry Kalvar - Brian Piccolo IS 53, Far Rockaway, New York
Stanley Kushel - Shell Bank IS 14, Brooklyn, New York
Fayvian Lee - Computer and Information Sciences Unit
Saundra Newman - Stuyvesant High School, New York, New York
Baruch Rapaport - Computer and Information Sciences Unit
Richard Rothenberg - Stuyvesant High School, New York, New York

The material contained in the final version of this guide was reviewed by Gerald Elgarten, Director of Mathematics (Interim Acting); Randy Ross, Principal, Newtown High School, the following Assistant Principals (Supervision - Mathematics): Pierre Fabinski (Christopher Columbus H.S.), Joel Friedberg (Herbert Lehman H.S.), Marc Licht (Computer and Information Sciences Unit), Demosthenes Lyriotakis (Manhattan Voc.-Tech. H.S.), Mary Gahn (John Bowne H.S.), Alice Farkouh (Curtis H.S.), Robert Frankel (Tottenville H.S.), Maryann Marrapodi, and members of the Computer and Information Sciences Unit, namely, Neal Ehrenberg, Fred Goldberg and Wilson Valcin.

The manuscript was prepared for publication by the Editorial and Production Unit, under the direction of Alaire Mitchell, Assistant Director of Curriculum Research (Acting).

Peter Dews and Vincent Puglia edited the document for the Editorial and Production Unit, and Vincent Puglia was responsible for the production.

Graphics supervision: Maureen Maguire
Word processing: Gloria Solomon
Cover: Kent Beaty

TABLE OF CONTENTS

SECTION 1: ADVANCED PROGRAMMING CONCEPTS.....1

TOPIC 1
SCREEN FORMATTING.....2

TOPIC 2
GRAPHICS ON THE IBM.....11

TOPIC 3
COMMODORE LO-RESOLUTION GRAPHICS.....16

TOPIC 4
COMMODORE HI-RESOLUTION GRAPHICS.....20

TOPIC 5
APPLE GRAPHICS MODE.....25

TOPIC 6
APPLE HI-RESOLUTION GRAPHICS -- PART I.....28

TOPIC 7
APPLE HI-RESOLUTION GRAPHICS -- PART II.....31

TOPIC 8
ANIMATION -- PART I.....35

TOPIC 9
ANIMATION -- PART II.....42

TOPIC 10
ANIMATION TO ENHANCE CAI -- PART I.....52

TOPIC 11
ANIMATION TO ENHANCE CAI -- PART II.....59

TOPIC 12
APPLE MEMORY LOCATIONS.....67

TOPIC 13
USING PEEK AND POKE ON AN APPLE COMPUTER.....70

TOPIC 14
USING SOUND AS A VEHICLE FOR INSTRUCTION.....73

TOPIC 15
SOUND GENERATION -- IBM.....80

TOPIC 16
SOUND GENERATION -- APPLE -- PART I.....84

TOPIC 17
SOUND GENERATION -- APPLE -- PART II.....87

TOPIC 18
INTRODUCTION TO APPLE SEQUENTIAL TEXT FILES.....90

TOPIC 19
COMMODORE TAPE AND DISK SEQUENTIAL INPUT COMMANDS.....94

TOPIC 20
IBM PC AND TRS-80 DISK SEQUENTIAL FILES.....97

TOPIC 21
MANAGING APPLE TEXT FILES.....101

TOPIC 22
APPLE RANDOM TEXT FILES.....104

TOPIC 23
IBM PC AND TRS-80 RANDOM ACCESS FILES.....108

TOPIC 24
TRS-80 CASSETTE FILES.....112

SECTION 2: SPREADSHEET MANAGEMENT.....116

TOPIC 25
INTRODUCTION TO VISICALC.....117

TOPIC 26
USING VISICALC FORMULAS AND COMMANDS.....120

TOPIC 27
VISICALC AND BASIC PROGRAMMING -- PART I.....124

TOPIC 28
VISICALC AND BASIC PROGRAMMING -- PART II.....128

SECTION 3: TELECOMMUNICATIONS.....135

TOPIC 29
ON-LINE DATA BASES.....136

TOPIC 30
BASIC TELECOMMUNICATIONS -- PART I.....140

TOPIC 31
TELECOMMUNICATIONS -- PART II.....143

TOPIC 32
TELECOMMUNICATIONS: MODEMS -- PART III.....146

TOPIC 33
TELECOMMUNICATIONS -- PART IV.....150

TOPIC 34
DATABASE INFORMATION SERVICES -- PART I.....153

TOPIC 35
DATABASE INFORMATION SERVICES -- PART II.....157

SECTION 4: THE MAJOR PROJECT.....161

TOPIC 36
COMPUTER ASSISTED INSTRUCTION -- WHAT IS IT?.....162

TOPIC 37
PLANNING A CAI PROJECT.....164

TOPIC 38
SCREEN FORMAT.....167

TOPIC 39
ERROR TRAPPING.....169

TOPIC 40
CREATING A MULTIPLE CHOICE TEST.....171

TOPIC 41
EVALUATING THE PROJECT.....174

SECTION 5: COMPUTERS IN SOCIETY.....176

TOPIC 42
MEDICAL DIAGNOSIS.....177

TOPIC 43
LABORATORY TESTING.....182

TOPIC 44
BIONICS.....187

TOPIC 45
COMPUTERS IN THE LEGAL PROFESSION.....192

TOPIC 46
COMPUTER SIMULATION.....196

TOPIC 47
DATABASES AND SECURITY ISSUES.....201

TOPIC 48
DEPERSONALIZATION IN THE ELECTRONIC AGE.....206

SECTION 6: OTHER NUMBER SYSTEMS 208

TOPIC 49
BINARY/DECIMAL CONVERSIONS 209

TOPIC 50
HEXADECIMAL/BINARY CONVERSIONS 214

TOPIC 51
NUMBER BASE SYSTEMS 218

SECTION 7: PROGRAMMING WITHOUT BASIC 222

TOPIC 52
COMPILERS 223

TOPIC 53
THEORY OF LANGUAGES -- MACHINE LANGUAGE 226

TOPIC 54
THEORY OF LANGUAGES -- ASSEMBLY LANGUAGE 229

TOPIC 55
THEORY OF LANGUAGES -- HIGHER LEVEL LANGUAGES 231

TOPIC 56
AN INTRODUCTION TO LOGO
PRIMITIVE COMMANDS AND TURTLE GRAPHICS 233

TOPIC 57
EDITOR PROCEDURES IN LOGO 242

TOPIC 58
PERIPHERAL COMMANDS AND ADVANCED PROGRAMMING IN LOGO 251

TOPIC 59
AN INTRODUCTION TO PASCAL 259

TOPIC 60
CONTROL STRUCTURES IN PASCAL 263

TOPIC 61
DATA STRUCTURES IN PASCAL 266

SECTION 8: SOME USEFUL MATHEMATICS TOPICS 269

TOPIC 62
STATISTICS -- MEASURES OF CENTRAL TENDENCY 270

TOPIC 63
STATISTICS -- MEASURES OF DISPERSION 273

TOPIC 64
SOLVING QUADRATIC EQUATIONS..... 275

TOPIC 65
SOLVING POLYNOMIAL EQUATIONS..... 277

TOPIC 66
THE MONTE CARLO METHOD..... 279

SECTION 9: APPENDICES..... 281

APPENDIX A
COMPUTERS AND SPECIAL EDUCATION STUDENTS..... 282

APPENDIX B
SUGGESTIONS FOR FURTHER READING..... 285

APPENDIX C
GLOSSARY..... 289

APPENDIX D
EDUCATIONAL SOFTWARE VENDORS..... 292

APPENDIX E
COMPUTING MAGAZINES..... 300

APPENDIX F
POPULAR HARDWARE VENDORS..... 302

APPENDIX G
PRIMARY SOURCES FOR GRANT INFORMATION..... 303

NOTES ON THE USE OF THIS GUIDE

This book is the second part of a two volume series. The series was designed to provide teachers with enough support material to enable them to teach an advanced course in computer literacy. It is intended for teachers whose classes have completed all of the work presented in our earlier programming guides. Volume II extends the objectives of Volume I.

The topics in these books are each designed to encompass an entire unit of instruction rather than a specific time period. Some may require only a few minutes to present while others will take a few days to complete.

The average high school class studying this material for three lecture and two lab periods per week, should be able to successfully complete all of the topics in approximately three semesters.

The appendix section in Volume I contains eleven review topics. It was designed to assist teachers whose students have not studied computer literacy for more than one year, to refresh their weakened skills or as a review for more advanced students. The appendix sections included in Volume II include material to help teachers meet the needs of special education students, and those with needs for more advanced material.

Since each classroom situation is different, it is the staff of each school that must determine how this guide can be used most effectively. In fact, the individual lessons may be used in their current format or may be adapted to meet the particular needs of various groups of students.

INTRODUCTION

The use of computers in classrooms all over the United States has been little short of spectacular in recent years. Surveys taken late in 1983 indicate that more than 85% of all schools were using microcomputers as instructional tools. The use of microcomputers in education has grown at an exponential rate since their introduction in the late 1970's. The optimum method for harnessing the power of these tools, however is still a subject of considerable debate. The real issue is that there is no consensus on what educational computing should be.

Recent reports show that administrators and teachers feel comfortable discussing reading, math and other more familiar curricular problems. They have had years of experience and research to reinforce any decisions made to meet these challenges. Computer education is a very different matter. The technology is new and it is changing daily as is the support structure surrounding all of the new equipment. Software, texts, self-help guides and a vast array of peripheral equipment are regularly changing the face of educational computing. For one of the few times in recent history, educators across the country are faced with a common challenge: how to use this exciting and powerful technology in a way which will meet the changing needs of students and society within the constraints of existing computer technology and which will continue to accommodate these changes as they occur. The challenge is a great one, but the potential for these tools is even greater.

We know that computers ignite the imagination and can extend the limits of human learning. They enable their users to view the world in new ways. Research indicates that when students interact with computers they take charge of their own learning and learn more through this interaction. The problem of time on task disappears as students work with computers to solve problems in various ways. In short, computers are tools used by humans for human endeavors. The days of computers being used solely for "number crunching" are long past. Computers are now more accurately viewed as information processors, a much more factually correct and less limiting view.

The New York City Board of Education has devised a plan for implementing computer education in all of its schools and for all of its students so as to best meet this great challenge. After numerous meetings with representatives of all levels of our instructional system and related community agencies, the master plan was devised. This two volume Computer Science Curriculum Guide is but a small part of that plan. It has been designed as a support document for teachers whose students have already completed computer courses which appear earlier in the plan's sequence. We find ourselves at the dawn of a new era in education. We fervently approach the prospect and reach out to its promise.

Section

1

ADVANCED PROGRAMMING CONCEPTS

Topic 1
SCREEN FORMATTING

Performance Objectives: Students will be able to:

- learn how to decorate a text screen
- select an appropriate decoration for an introductory message to a program
- create "pretty screens" during the execution of a program
- organize screen layout to include input areas, information areas and regions for decorations
- learn technique for limiting input area to specific screen locations
- format the entry of specific types of data such as dates
- understand the meaning of windows on a text screen
- end program with an appropriate message placed on a "pretty screen"

Vocabulary: New: pretty screens, screen windows
Review: PRINT TAB(), GOSUB/RETURN, ASCII CODES, CHR$()

Motivation:

- What makes a computer text screen pretty to look at? (Interesting border, neat and orderly format, corner designs, etc.)

- What tools do we have available to us for the purpose of decorating a text screen? (A drawing area that consists of a given number of screen columns and a set number of screen lines. There are keys that draw letters, numbers, punctuation marks, and a variety of other keys depending on the system that you are using. We have normal, inverse and flashing modes, but we have no color. We shall attempt to use what we have to the best of our ability to create something that is both pleasing to look at and fun to work with.)

- Can different keystrokes be combined to create interesting designs? ("-+-+-+-" and "<>><>><>" and "(*)(*)(*)" are some of the ways of creating borders and lines of division.)

- What is the purpose of drawing these lines of division on the text screen? (The creation of "text windows.")

- What are text windows? (Portions of the screen that are designated for specific purposes.)

Aim: What are some of the ways that we can provide for "pretty screens" throughout a program?

Development:

1. In the past, how have we created borders that outlined our text screen?

   - Selected the specific keystrokes that created the border.

   - Determined size of text screen, C columns and R rows.

   - Top horizontal line created with a FOR/NEXT loop. The character(s) printed with a PRINT";" instruction.

- Leftmost column filled by a FOR/NEXT loop and a simple PRINT instruction.

- Bottom row follows the pattern of the top row.

- Rightmost column involves the use of cursor control keys. This varies from computer to computer and it is strongly advised that this technique be reviewed before any attempt at screen decoration is made. A thorough understanding of cursor positioning is vital to the success of this lesson. Once these controls have been reviewed we simply position the cursor at the top of the screen, in the last column, and proceed as we did on the left.

  - Various borders should be created with pleasing introductory messages, centered within.

  - Experiment with groups of keystrokes to be printed in the four corners. ° or _O_
    < > O

  - These can be defined as strings and printed anywhere. For example:

    A$ = " ° ":B$ = ">"
    Position the cursor at the desired location and PRINT A$: PRINT B$: PRINT C$

2. Develop with students criteria for a good introductory display.

   - Have students understand the importance of a clear introduction. We attempt to make the display "pretty" but we must avoid clutter or anything else that might be confusing to the user.

   - We set the tone for the program in our introduction and we must aim for clarity and simplicity.

   - Wide empty spaces add to simplicity. Therefore, we avoid surrounding every word with stars. On the other hand, if we wish to emphasize one word or phrase, surrounding it with stars would be a good idea.

   - Continue in this manner, showing the students sample displays and let them suggest improvements.

3. Numbering screen lines is an effective way of directing the placement of answers. Review the ASCII codes with students and the CHR$() instruction. Show a quick way of numbering screen lines. The ASCII codes for the digits and the letters are sequential. Numbers 0 through 9 are represented by codes 48 through 57. The letters A through Z are represented by 65 through 90.

   - Write a program that will create a window to occupy the bottom 12 screen lines. Then print the numbers 1 through 5 on every other screen line.

```
5   REM PROGRAM FOR A 40 COLUMN SCREEN
10  HOME: REM Clear screen
20  REM POSITION CURSOR TO TOP OF WINDOW
```

```
30  FOR X = 1 TO 10: PRINT : NEXT X
35  REM PRINT DIVIDING LINE
40  FOR X = 1 TO 8
50  PRINT "{-*-}";
60  NEXT X
65  REM NUMBER WINDOW LINES
70  A = 1:PRINT
80  FOR C = 1 TO 5
90  PRINT  CHR$ (A + 48);".": PRINT :A = A + 1: NEXT C
100 END
```

4. Pose problem: We wish to provide for the input of a date. It is necessary that this date be entered in the following format: A two-digit month, a two-digit day, and a two-digit year. What are some of the ways that we can explain to the user what we want?

   - Have students describe various messages explaining what we want to the user.
   - Teach students the method of forcing user input to a specific area of the screen.
   - Have them appreciate the simplicity of using "no words."
   - Review GOSUB/RETURN.
   - Explain the advantage of adapting this routine to many other situations.
   - Select the inverse mode to capture the user's attention.
   - Suggested program:

```
10  REM  THIS STEP TO CLEAR SCREEN
20  FOR X = 1 TO 10
30  PRINT : NEXT X
40  PRINT  TAB( 11)"PLEASE ENTER DATE"
42  PRINT : PRINT : PRINT
45  PRINT  TAB( 16);
50  GOSUB 500
60  PRINT "/";
70  GOSUB 500
80  PRINT "/";
90  GOSUB 500
100  REM  POSITION CURSOR AT THIS STEP ON THE 15TH SCREEN LINE AND
     THE 16TH COLUMN ACCORDING TO THE STATEMENTS AVAILABLE WITH YOUR
     MACHINE
110  REM  THIS STEP TO RECEIVE THE INPUT VIA INKEY$ OR GET STATEMENT
480  END
500  REM THIS SECTION PRINTS THE INPUT BOX
510  FOR A = 1 TO 2
520  PRINT " ";: NEXT A
530  RETURN
```

5. Providing for "pretty screens" throughout the program is an essential part of a user-friendly program. Varying the vertical and horizontal

format lends interest to a program. An example of a vertical format follows in a program that tests Spanish vocabulary.

```
IMPROVE YOUR SPANISH      *****
                          *   *
TRANSLATE EACH WORD.      *****

ESPANOL              INGLES
                *
1.  APIO        * 1.
                *
2.  CANASTA     * 2.
                *
3.  DESIDIA     * 3.
                *
4.  HAMBRE      * 4.
                *
5.  MANCEBO     * 5.
                *
6.  PROBETA     * 6.
                *
7.  RELUMBRANT  * 7.
                *
8.  SUSURRAR    * 8.
```

- Note the different windows. We will keep score in the upper right hand corner of the screen. The right half is used for questions and the left half for answers. The top is used for titles and simple instructions.

- Make use of the ASCII codes for writing the numbers.

- Adjust screen display to suit your particular computer. This is formatted for a 40 column display.

- Suggested program:

```
3   REM SPANISH VOCABULARY PRACTICE
5   REM DEFINE STRINGS TO CREATE SCORE BOX
10  A$ = "*****"
20  B$ = "*   *"
30  REM Clear screen using a version available on your machine
40   PRINT  TAB( 7)"IMPROVE YOUR SPANISH" TAB( 34)A$
50   PRINT  TAB( 34)B$
60   PRINT  TAB( 7)"TRANSLATE EACH WORD." TAB( 34)A$
70   PRINT : PRINT
80   PRINT "ESPANOL" TAB( 22)"INGLES"
90   PRINT : PRINT "1.  APIO"
100   PRINT : PRINT "2.  CANASTA"
110   PRINT : PRINT "3.  DESIDIA"
120   PRINT : PRINT "4.  HAMBRE"
130   PRINT : PRINT "5.  MANCEBO"
140   PRINT : PRINT "6.  PROBETA"
150   PRINT : PRINT "7.   RELUMBRANT"
160   PRINT : PRINT "8.  SUSURRAR"
170   REM  THE FOLLOWING SECTION OF THE PROGRAM IS INTENDED FOR
```

```
     COMPUTERS THAT SUPPORT VTAB AND HTAB
190  REM POSITION CURSOR PRINT VERTICAL LINE OF STARS
200  VTAB 5: FOR T = 1 TO 18: HTAB 20: PRINT "*": NEXT
205  REM POSITION CURSOR PRINT NUMBERS
210   V = 8:A = 1
215  FOR X = 1 TO 8
220     VTAB V: HTAB 22: PRINT  CHR$ (A + 48)"."
230     A = A + 1:V = V + 2
240  NEXT X
```

If your machine does not support VTAB and HTAB then you must select equivalent statements which will work for you.

6. Programs that lend themselves to horizontal formats are ones that include information that is necessary for the solution of a problem. Tables or involved instructions are examples of programs suited for horizontal formats. One such program is the following chemistry lesson. We wish to format and decorate our screen in the following manner:

```
                    DENSITY OF SUBSTANCES
                    =====================

SUBSTANCE           G./CC.  LBS./CU.FT.
---------           -----   -----------
CORK                 0.22       13.7
GOLD                19.3      1204.3
ICE                  0.197      57.2
MERCURY             13.6       849.9
WATER, FRESH         1.0        62.4
ZINC                 7.1       443.0

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    REFER TO THE ABOVE TABLE TO
   ANSWER THE FOLLOWING QUESTION
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

>WHICH SUBSTANCES WOULD FLOAT ON WATER?

ANS.

>WHICH SUBSTANCES WOULD SINK IN MERCURY?

ANS.
```

- Discuss the different windows.
- Point to the alignment of the decimal point and agree to accomplish this by adjusting the PRINT TAB() statement.
- Point out the spaces left for the answers to the questions.
- Review the use of the semicolon after the PRINT TAB().
- Suggested program:

```
5   REM CHEMISTRY PRACTICE
9   REM Clear the screen
10   PRINT  TAB( 10)"DENSITY OF SUBSTANCES"
15   PRINT  TAB( 10);: FOR X = 1 TO 20: PRINT "=";: NEXT X
20   PRINT : PRINT  TAB( 3)"SUBSTANCE" TAB( 20)"G./CC."
     TAB( 29)"LBS./CU.FT."
25   PRINT  TAB( 3)"---------" TAB( 20)"------" TAB( 29)"-----------"
30   PRINT  TAB( 3)"CORK" TAB( 21)"0.22" TAB( 31)"13.7"
40   PRINT  TAB( 3)"GOLD" TAB( 20)"19.3" TAB( 29)"1204.3"
50   PRINT  TAB( 3)"ICE" TAB( 21)"0.197" TAB( 31)"57.2"
60   PRINT  TAB( 3)"MERCURY" TAB( 20)"13.6" TAB( 30)"849.9"
70   PRINT  TAB( 3)"WATER,FRESH" TAB( 21)"1.0" TAB( 31)"62.4"
80   PRINT  TAB( 3)"ZINC" TAB( 21)"7.1" TAB( 30)"443.0"
85   PRINT
90   FOR X = 1 TO 40: PRINT "%";: NEXT X
100   PRINT  TAB( 8)"REFER TO THE ABOVE TABLE TO"
110   PRINT  TAB( 7)"ANSWER THE FOLLOWING QUESTION"
120   FOR X = 1 TO 40: PRINT "%";: NEXT X
125   PRINT
140   PRINT ">WHICH SUBSTANCES WOULD FLOAT ON WATER?"
150   PRINT "ANS.": PRINT
160   PRINT ">WHICH SUBSTANCES WOULD SINK IN MERCURY?";
165   PRINT "ANS."
170   REM POSITION CURSOR ON 18th SCREEN LINE AND 5th COLUMN
175   REM ACCEPT INPUT
180   REM POSITION CURSOR ON 21st SCREEN LINE AND 5th COLUMN
190   REM ACCEPT INPUT
```

7. What are some of the reasons why we must inform the user that a program has ended?

   - If the screen goes blank and nothing happens the user might think that the computer is broken.

   - Frantically pressing keys in an attempt to get a reaction from the computer can have a negative effect on both the user and the computer.

   - Our aim is to leave the user with a pleasant feeling, one of accomplishment and satisfaction.

     - Develop a list of pleasant sign off messages. Include words of encouragement and messages which anticipate future meetings. Discuss appropriate screen decorations.

     - Some samples:

```
     *****
    * ^ ^ *
    ( o o )
    *  v  *
   ** |-/ **

HAVE A NICE DAY
```

```
(*)(*)(*)(*)(*)(*)(*)(*)(*)(*)(*)(*)(*)
(*)                                 (*)
(*)       Y  O  U     D  I  D       (*)
(*)                                 (*)
(*)    V  E  R  Y     W  E  L  L    (*)
(*)                                 (*)
(*)(*)(*)(*)(*)(*)(*)(*)(*)(*)(*)(*)(*)
```

Applications:

1. Design an introduction to a program that provides review for the SAT exam. Include the following in your program:

   . a screen border

   . The words SAT REVIEW

   . Emphasize SAT in some way

2. Use the CHR$() instruction in conjunction with the ASCII codes to create the following display.

   . Use strings to simplify the printing of the design.

   . Use inverse mode for "select letter."

   . Use flashing mode for arrow.

   . Position cursor inside the "< >" to await user input.

```
 O          GRAPHICS MENU          O
O*O                              O*O
 O                 < A >          O
 !                                !
|!/                < B >         |!/
 O                                O
O*O                < C >         O*O
 O                                O
 !                 < D >          !
|!/                              |!/
 O                 < E >          O
O*O                              O*O
 O                 <F>            O
 !                                !
|!/                <G>           |!/
 O                                O
O*O                <H>           O*O
 O                                O
 !                 <I>            !
|!/                              |!/

   SELECT LETTER -->< >
```

3. Write a program that asks the user to input the high school's name.

   . Create a screen window composed of 20 inverse horizontal blanks.

- Have the computer add the phrase "high school."
- Use an appropriate screen decoration (suggestion follows).

```
  _
 !_!
 !
XXX
XOX
XXXXXXX
X O O X
X O O X
XXXXXXX
```

Summary:

1. When decorating a screen is it always necessary to draw a picture?

2. Do introductory and terminal screen messages lend themselves to different types of screen decorations?

3. What type of a screen decoration would be used in the body of a program?

4. What is the purpose of using a screen window?

5. What is a fast way of labeling every other screen line with the letters A through L?

6. What are the commands we use to position the cursor?

7. How do "pretty screens" affect a user friendly program?

Assignments:

1. Design a sign-off program that accomplishes the following:

   - Informs the user that a score of 86 is very good.

   - Encourages the user to try for 100 the next time.

   - Includes some type of screen decoration.

2. Assume that you are using a 40-column screen. Display the output for each of the following instructions:

   - 10 FOR X = 1 TO 10:PRINT TAB(40)"X";:NEXT X

   - 10 FOR X = 1 TO 10:PRINT TAB(40)"X":NEXT X

3. Write a program that tests a user's skill at identifying authors. Include the following in your program:

   - Use a vertical format.

   - Name the program "Great Works of Literature."

   - Include the names of 5 books.

- Provide for a score box.
- Use a question window.
- Include an answer window.
- End the program with the cursor positioned to accept the first answer.

Assignment for More Advanced Students:

1. Complete the program in exercise 3.
   - Allocate an appropriate point value for each correct answer.
   - Display the changing score in the score window.
   - Test each answer as it is input.
   - Give the user two chances to find the correct answer before you give it away.
   - Have the cursor jump to an appropriate position to await the next input.

Topic 2

GRAPHICS ON THE IBM

Performance Objectives: Students will be able to:

- understand the difference between medium and high resolution
- understand the choices when generating color on the screen
- use the PSET and PRESET statements to generate diagrams using lines

Vocabulary: New: resolution, pixel, foreground, background, palette, COLOR, KEY OFF, SCREEN, PSET, PRESET, LINE, CIRCLE

Motivation:

- How would you get the computer to generate a picture of a rectangle on the screen?

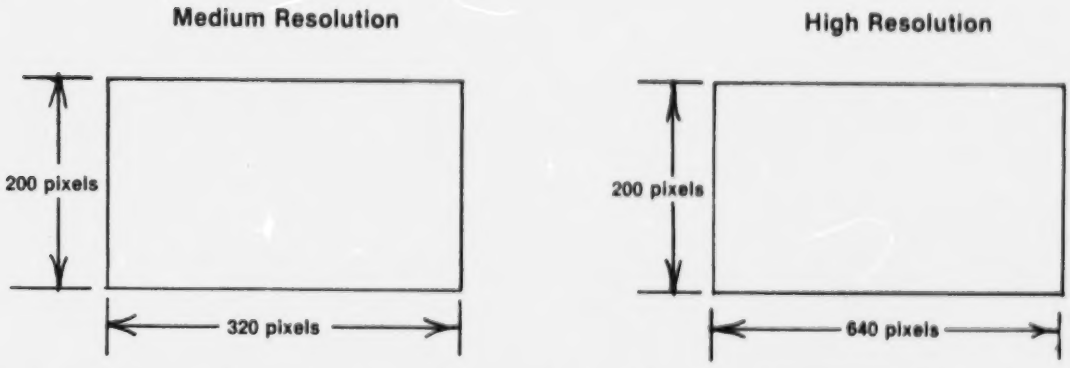
Aim: How do you set up the screen for graphics? What do the statements COLOR, KEY OFF, SCREEN, PSET AND PRESET do?

Development:

1. Explain that there are a number of graphics statements that can be used with the IBM PC but some can be used only with DOS BASIC and, in some cases, only with Advanced DOS BASIC (BASICA).

2. Explain that the screen can be considered a grid of points known as pixels. In medium resolution, there are 320 horizontal pixels and 200 vertical pixels. In high resolution, there are 640 horizontal pixels and 200 vertical pixels.

Diagram 1



3. To prepare the screen for the graphics mode, the SCREEN statement is used.

```
100 SCREEN 1, 0
```

This will cause the computer to prepare for the color graphics mode in medium resolution.

BEST COPY AVAILABLE

4. There are sixteen colors that can be generated on the PC. Four colors can be generated at once in medium resolution. In high resolution, two colors can be generated. In medium resolution, one color is used for background and three colors are possible for the foreground. The background color can be any of sixteen colors. These are:

0	black	8	gray
1	blue	9	light blue
2	green	10	light green
3	cyan	11	light cyan
4	red	12	light red
5	magenta	13	light magenta
6	brown	14	yellow
7	white	15	high-intensity yellow

5. The COLOR statement is used to select the colors used. The general format of the COLOR statement is:

```
COLOR background, palette
```

where the color selection for each palette is as follows:

Color	Palette 0	Palette 1
1	green	cyan
2	red	magenta
3	brown	white

For example:

```
215 COLOR 10,1
```

sets the background to light green and chooses palette 1. What statement would you write to set the background to magenta and to select palette 0? (102 COLOR 5,0)

6. The KEY OFF statement erases the listing of the function keys on the last line of the display.

7. The PSET and PRESET statements both put a point of a specified color at a particular location on the screen. The difference between the two statements is that if no color is specified for the PRESET statement, the background color will be black (0). For the PSET statement, if the color is not specified, the background color will be white. The general form of the PSET and PRESET statements is:

```
PSET (x,y), color
PRESET (x,y), color
```

where x can be a number from 0 to 319 and y a number from 0 to 199, and the color is a number from 0 to 3 from the specified palette.

8. Examine the following program:

```
10  SCREEN 1,0
20  KEY OFF
30  FOR I=100 TO 150
40      PSET (50,I),2
50      PSET (100,I),2
60  NEXT I
70  FOR I=50 TO 100
80     PSET (I,100),2
90      PSET (I,150),2
100 NEXT I
110 END
```

- What does this program do in statement 10? 20? 30-60? 70-100?

- How would you modify the program so that the rectangle would be blue?

- How would you modify the program using the COLOR statement to make the background white and the rectangle magenta?

9. Explain that the LINE statement is another way to draw lines that is easier and faster than the PSET and PRESET statements. A simple general format of the line statement is:

```
110  LINE (x1,y1)-(x2,y2),color
```

where (x1,y1) are the coordinates of the starting point of the line and (x2,y2) are the coordinates of the end of the line.

- For example, examine the following program:

```
10 REM DRAWING A TRIANGLE
20 CLS
30 KEY OFF
40 COLOR 3,0
50 LINE(50,100)-(100,50),2
60 LINE(100,50)-(150,100),2
70 LINE(150,100)-(50,100),2
80 END
```

  - How would you modify this program to change the color of the triangle to brown? magenta?

  - How would you modify this program to set the background to yellow and the triangle to green?

- The following example is an illustration of how the LINE statement can be used to draw rectangles:

```
10 CLS
20 KEY OFF
30 SCREEN 1,0
40 COLOR 14,0
50 LINE(100,50)-(200,75),L,B
60 END
```

where (100,50) are the coordinates of the top left corner of the rectangle and (200,75) are the coordinates of the bottom right corner of the rectangle. There is also a feature in which the rectangle can be filled with a specific color. Examine the following:

```
50 LINE(100,50) -(200,75),1,BF
```

- What is the difference in output if this line 50 replaces the line 50 in the above program?

10. Explain that the CIRCLE statement can be used only in Advanced BASIC. The simplest form of this statement is:

```
10 CIRCLE(x,y),r,color
```

where (x,y) are the coordinates of the center of the circle and r is the radius. For example, examine the following program:

```
10 CLS
20 SCREEN 1,0
30 KEY OFF
40 COLOR 6,1
50 CIRCLE(125,100),25,2
60 END
```

- In statement 50, what does the 25 represent? (125,100)? 2?
- How would you modify this program to draw a red circle?
- How would you modify this program to draw a blue circle on a white background?

Applications:

1. Write a program that will draw five concentric circles.
2. Write a program that will draw the following diagram:

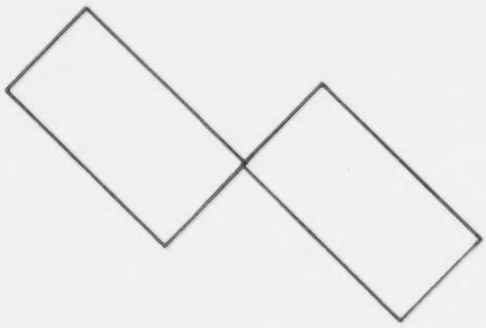


Diagram 2

Summary:

1. What is the difference between medium and high resolution?

2. What is the difference between the PSET and PRESET statements?

3. What is the advantage of using the LINE statement over the PSET or PRESET statements?

Assignments:

1. Write a program that will draw a filled-in red rectangle on a light blue background.

2. Write a program that will draw a blue triangle on a yellow background that blinks on and off.

3. Write a program that will draw a hexagon.

4. Write a program that will draw a set of three circles which intersect each other.

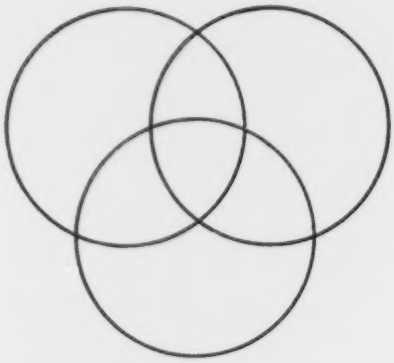


Diagram 3

Topic 3

COMMODORE LO-RESOLUTION GRAPHICS

Performance Objectives: Students will be able to:

- use keyboard graphics characters
- create original graphics by concatenating strings
- use a subroutine to PRINT a string anywhere on the screen
- create programs that can construct bar graphs

Vocabulary: New: lores, character string, character set, concatenate, graphics characters

Motivation:

- Study the keyboard of the Commodore 64. What symbols are on the keys? (letters, numbers, punctuation marks, control keys such RETURN and SHIFT, abbreviations for colors, cursor arrows, function numbers, mathematical symbols, and those funny little drawings on the front of each key)

- Because these drawings and symbols can be used to draw or illustrate on the screen, they are called graphics characters.

Aim: How can we use the built-in graphics characters on the Commodore 64?

Development:

1. Describe the graphics characters on the front of the keys. Can they be grouped into categories?

   - Some are easily describable: the suits of a deck of cards, open and closed circles, checkerboard patterns. Others are more abstract: lines of varying heights and thicknesses (some vertical, some horizontal), angles, and corners with various orientations.

2. Can you find patterns of arrangement on the keyboard? (Notice the cluster of corners on the A, S, Z, and X keys. Are there other such arrangements?)

3. How can these graphics be displayed on the screen? (The design on the right side of the key can be obtained by pressing that key while depressing the SHIFT key. The design on the left side of the key is obtained by pressing the key while depressing the Commodore key (the one on the lower left of the keyboard with the Commodore logo on it). We will symbolize that as <CMDR>.)

4. Experiment by writing PRINT statements using the various characters.

5. Notice that printing several of these characters together creates a new arrangement. What are several ways we can create these new strings?

   - Use graphics characters in a literal string.

Example:

PRINT "<SHIFT>0<SHIFT>P" gives a half-box.

- Cursor controls can be used within the string as well.

  Example:

  PRINT "<SHIFT>0<SHIFT>P<CRSR DOWN><CRSR LEFT><CRSR LEFT><SHIFT>L <SHIFT>@" will give a box over two screen lines.

- Assign such values to string variables with LET statements.

  Example:

  LET A$ = "<SHIFT>W"

  Whenever A$ is PRINTed, an open circle will appear.

- Concatenation of strings. By adding strings, we can create new strings.

  Example:

```
10 LET A$ = "<CMDR>A":    REM A CORNER
20 LET B$ = "<CMDR>S":    REM ANOTHER CORNER
30 LET C$ = A$+B$:        REM CONCATENATE THE CORNERS
40 PRINT C$:              REM DISPLAYS NEW DESIGN
```

- Reverse keys and color control keys make the possibilities endless. Notice the effect that REVERSE has on the space bar. This can be used to create solid bars of color.

- Experiment freely. Try to create spaceships, stick figures of people or animals, outlines for charts or business-type forms.

6. How can we get a graphics character directly from ROM?

- Each character available from the keyboard (letters, digits or graphics) is in the computer's permanent memory (ROM). There they are called character strings and are each assigned a number from 0 to 255. The entire list is referred to as the computer's character set and varies from one type of computer to another.

- They can be accessed directly through the CHR$ function (read CHR$ as "character string"). This function is usually used with either PRINT or LET commands.

  Examples:

  PRINT CHR$(115) will display a heart.

  Or:

```
10 LET A$ = CHR$(115)
20 PRINT A$
```

  will also print a heart.

- These character strings can be concatenated.

  Example:

```
30 LET B$ = A$+CHR$(120)+CHR$(122)+CHR$(129)
40 PRINT B$
```

  This will display the heart, club, diamond and spade.

  (Note: A drawing of each character and its code number may be found on pp. 135-137 of the Commodore 64 User's Guide and on pp. 379-381 of the Commodore 64 Programmer's Reference Guide.)

7. How can we PRINT a character string anywhere on the screen?

   - BASIC on the Commodore 64 does not include a PRINT @ command which would place the output at a chosen spot on the screen. The following routine will substitute for that missing command:

```
1000 REM CURSOR POSITION TO ROW AND COLUMN
1010 PRINT "<CLR>"
1020 IF ROW = 2 THEN 1060
1030 FOR I = 1 TO ROW-1
1040 PRINT
1050 NEXT
1060 PRINT SPC(COL);
1070 RETURN
```

   - Note that the variables ROW (the row down, an integer from 1 to 25) and COL (for column across, an integer from 0 to 39) must have values assigned to them before the subroutine is used. For instance, complete the program with the following:

```
100 ROW = 15:COL = 10
110 INPUT"YOUR NAME PLEASE";N$
120 GOSUB 1000
130 PRINT N$
140 END
```

Summary:

1. Substitute the following lines and note the result:

```
10 PRINT "<CLR>"
20 ROW = 10:COL=1:GOSUB 1000
30 FOR J = 1 TO 40
40     PRINT "<CMDR> @";: REM This command could also be PRINT CHR$(164)
50 NEXT
60 END
```

   - This should print a horizontal line ten lines down the screen. Try this with different colors and with the REVERSE on or off.

Assignments:

1. Modify this routine so that it will draw horizontal lines or bars of any length from 0 to 40.

2. Expand the program so that it will draw several lines or bars of varying lengths.

3. Complete the program by adding input and scaling routines so that it will generate a horizontal bar graph.

Assignment for More Advanced Students:

1. Write a routine that will draw a vertical bar graph.

Topic 4

COMMODORE HI-RESOLUTION GRAPHICS

Performance Objectives: Students will be able to:

- appreciate complexity of hi-resolution computer graphics
- see how subroutines are used to simplify programming
- understand importance of high level commands to access special functions of a machine
- have a basis for writing their own graphics programs

Vocabulary: New: hires, pixels, bit map, sine-wave
Review: lores

Motivation:

- Show class very hi-resolution graphics pictures as used in magazine ads, computer magazines, or stills from movies such as TRON or STAR WARS.

  - How do these pictures differ from drawings using the characters built into the computer?

  - How can we begin to create such images on the Commodore 64?

Aim: How are hi-resolution graphics created on the Commodore 64?

Development:

1. What is the difference between low resolution and high resolution (lores and hires) graphics?

   - Low resolution - you must use the character set built into the computer's ROM.

   - High resolution - you can control every dot on the screen individually. Each dot, therefore, becomes a single element of the finished picture and is called a picture element or pixel.

2. What method must we follow to be able to control all the pixels on the C-64 screen?

   - The C-64 has powerful graphics capabilities, but the version of BASIC that Commodore provides does not allow for its easy use. It should be pointed out that some machines, such as Apple, Radio Shack Color Computers, etc., have graphics commands, such as HIRES, PLOT, LINE, CIRCLE, which the C-64 does not. Commands such as these can be added to the C-64 by various extensions to BASIC, such as Simons' BASIC and Super Expander (Commodore Business Machines), Graphics BASIC (HES), Screen Graphics 64 and Ultrabasic 64 (Abacus Software), and others.

   - Show the students that the sample routines we will work with are difficult to understand, slow in operation, and can be achieved with a few commands using an enhanced BASIC. In preparation for using these routines, several tasks should be completed:

- Prepare a graphics screen.

- Calculate which pixels are to be ON or OFF.

- Turn on the proper pixels.

3. But first, a few general ideas. The screen can be thought of as a grid of points of light, each of which is either ON or OFF. In the computer's memory there is a digital map of this grid, called a bit map, telling the condition of each dot or pixel. On the C-64, the grid has 320 dots across (the X axis) and 200 up and down (the Y axis), making a grid of 320 x 200 or 64,000 dots. The dots are numbered as X, Y coordinates, starting at (0,0) at the upper left of the screen and moving to 319,199 at the bottom right.

4. Preparing a graphics screen:

   - The first program is really a subroutine that will be used in the other programs as lines 1000 - 1099. This is what the first program does: Lines 1010 to 1050 set up the hires mode, set aside memory locations 8192 to 16191 as a bit map, and places zeros in all those locations. The variable PIX indicates the color of the lit pixels (13 is light blue. BKGD sets the color of the OFF pixels, which form the background (6 is dark blue). You may change these to any number from 0 to 15. Some combinations work better than others. Have students experiment. All dots are then set to OFF (line 1070-1090). This routine takes 30-40 seconds.

6. Calculating which pixels should be turned on:

   - This is accomplished in various ways in these programs. Program 2 allows the user to INPUT the X,Y coordinates for a single point (line 100). Program 3 allows INPUT for the coordinates of two points (100-110) and then calculates the slope of the line between them and the coordinates of the points along that line (140-160 and 200-220). Program 4 calculates the vertical points along a sine wave crossing the screen (140-160).

7. Turning on the proper pixel:

   - Programs 2 and 3 use lines 2000-2040 to turn on a single point X,Y. Program 3 gets the value for the coordinate from the calculations. Program 4 uses the same subroutine but for two purposes. First, it plots the lines for each axis from X,Y information calculated in 3000-3080, and then to plot the actual graph from the X,Y values calculated in 140-150.

<u>Summary</u>:

1. These programs can be introduced all at once or developed in sections as they are presented below.

2. Points to emphasize:

   - the complexity of computer graphics
   - the need for higher level graphics routines or specialized programs

. the usefulness of modular programming and subroutines

Assignments:

1. Modify Program 2 to allow inputs of a user-chosen number of coordinates, and will plot them all.

2. Modify Program 3 to draw several lines or other geometric figures (rectangles, quadrilaterals, triangles, etc.) given the coordinates of their points or corners.

3. Modify Program 4 to draw other functions as well.

PROGRAM 1

```
1000 REM *** PREPARE SCREEN FOR
1005 REM *** HI RESOLUTION BIT MAP
1010 POKE 53265,PEEK (53265) OR 32
1020 POKE 53272,PEEK (53272) OR 8
1030 FOR I = 8192 TO 16191
1040    POKE I,O
1050 NEXT I
1060 PIX=13:BKGD=6
1070 FOR I = 1024 TO 2093
1080    POKE I,PIX*16+BKGD
1090 NEXT I
1099 RETURN
```

PROGRAM 2

```
10 REM *** PLOT A POINT X PIXELS ACROSS
20 REM *** AND Y PIXELS DOWN
30 REM*********************************
100 INPUT X,Y
110 GOSUB 1000
120 GOSUB 2000
130 END
1000 REM *** PREPARE SCREEN FOR
1005 REM *** HI RESOLUTION BIT MAP
1010 POKE 53265,PEEK (53265) OR 32
1020 POKE 53272,PEEK (53272) OR 8
1030 FOR I = 8191 TO 16191
1040     POKE I,0
1050 NEXT I
1060 PIX=13:BKGD=6
1070 FOR I = 1024 TO 2023
1080    POKE I,PIX*16+BKGD
1090 NEXT I
1099 RETURN
2000 REM *** POINT X,Y
2010 BY=8192+320+INT(Y/8)+8*INT(X/8)+(Y AND 7)
2020 BI=7-(X AND 7)
2030 POKE BY, PEEK(BY) OR 2↑BI
2040 RETURN
```

PROGRAM 3

NOTICE THAT LINES 1000-2040 ARE IDENTICAL TO THOSE IN PROGRAM 2

```
10 REM *** PLOT A LINE WHOSE ENDPOINTS
20 REM *** ARE X1,Y1 AND X2,Y2
30 REM ********************************
100 INPUT "ONE END POINT (X,Y)";X1,Y1
110 INPUT "OTHER END POINT (X,Y)";X2,Y2
120 GOSUB 1000
130 IF X1=X2 THEN GOTO 200
140 SL=(Y1-Y2)/(X1-X2)
150 FOR I = 0 TO (X2-X1) STEP SGN (Y2-Y1)
160    X=X1+I:Y=Y1+I*SL:GOSUB 2000
170 NEXT I
180 GOTO 180
190 END
200 FOR I = 0 TO (Y2-Y1) STEP SGN(Y2-Y1)
210    X=X1:Y=Y1+I:GOSUB 2000
220 NEXT I
230 GOTO 230
240 END
1000 REM *** PREPARE SCREEN FOR
1005 REM *** HI RESOLUTION BIT MAP
1010 POKE 53265,PEEK(53265) OR 32
1020 POKE 53272,PEEK(53272) OR 8
1030 FOR I = 8192 TO 16191
1040    POKE I,0
1050 NEXT I
1060 PIX=13:BKGD=6
1070 FOR I = 1024 TO 2023
1080    POKE I,PIX*16+BKGD
1090 NEXT I
1099 RETURN
2000 REM *** POINT X,Y
2010 BY=8192+320+INT(Y/8)+8*INT(X/8)+(Y AND 7)
2020 BI=7-(X AND 7)
2030 POKE BY, PEEK(BY) OR 2↑BI
2040 RETURN
```

PROGRAM 4

NOTICE THAT LINES 1000-2040 ARE IDENTICAL TO THOSE IN THE OTHER PROGRAMS

```
10 REM *** PLOT A FUNCTION
100 DEF FNF (X)=30*SIN(X)
110 A=6:B=40
120 GOSUB 1000
130 GOSUB 3000:REM *** DRAW AXES
140 FOR X - 0 TO 319
150     Y=100-(100/B)*FNF(A*(-160/159)
160     GOSUB 2000
170 NEXT X
180 END
```

```
1000 REM *** PREPARE SCREEN FOR
1005 REM *** HI RESOLUTION BIT MAP
1010 POKE 53265,PEEK(53265)OR 32
1020 POKE 53272,PEEK(53272) OR 8
1030 FOR I = 8192 TO 16191
1040    POKE I,0
1050 NEXT I
1060 PIX=13:BKGD=6
1070 FOR I = 1024 TO 2023
1080     POKE I,PIX*16+BKGD
1090 NEXT I
1099 RETURN
2000 REM *** POINT X,Y
2010 BY=8192+320+INT(Y/8)+8*INT(X/8)+(Y AND 7)
2020 BI=7-(X AND 7)
2030 POKE BY, PEEK(BY) OR 2↑BI
2040 RETURN
3000 Y=100
3010 FOR X = 0 TO 319
3020     GOSUB 2000
3030 NEXT X
3040 X=160
3050 FOR Y = 0 TO 199
3060     GOSUB 2000
3070 NEXT Y
3080 RETURN
```

Topic 5

APPLE GRAPHICS MODE

Performance Objectives: Students will be able to:

- understand the various graphics modes the Apple II can generate
- use PEEK and POKE statements in order to switch the various graphics modes

Vocabulary: Review: PEEK, POKE, HGR, HGR2

Motivation:

- Type in the following program. Before running it, however, can you predict what it will do?

```
5 REM SURPRISE
10  HCOLOR= 3
20  HOME : HGR  30  HPLOT 0,0 TO 279,0 TO 279,190 TO 0,190 TO 0,0
40  POKE  - 16300,0
50  POKE  - 16302,0
60  DIM X(21),Y(21): FOR T = 2 TO 21: X(T) = 78*SIN(.314*T)+140: Y(T) =
     78*COS(.314*T)+80
70  NEXT: FOR T = 2 TO 21: FOR Q = T TO 21: HPLOT X(T),Y(T) TO X(Q),Y(Q):
    NEXT: NEXT: VTAB 24
```

- Write your prediction before running this program. Then describe what the program did.

- Can you explain why things happened the way they did?

Aim: How can we use POKE and PEEK commands to control the various graphics modes?

Development:

1. How many graphics modes does the Apple have? (There are two basic modes: High Resolution and Low Resolution. However, High Resolution mode has two viewing screens, a page 1 and a page 2. In page 1, we can have a full screen of graphics, or a graphics screen with four text lines at the bottom.)

2. How can we control these various switches and modes?

   Acceptable answer:

   In BASIC we can use POKE statements to various locations. Certain memory locations are activated by simply POKEing a zero to a location. For example, POKE -16304,0 switches the Apple into graphics. The table below lists the memory locations and functions they control:

Memory Location	As a positive #	Function
-16304	49232	GRAPHICS
-16303	49233	TEXT
-16302	49234	FULL-GRAPHICS

-16301	49235	SPLIT-SCREEN
-16300	49236	PAGE 1
-16299	49237	PAGE 2
-16298	49238	LOW-RESOLUTION
-16297	49239	HIGH-RESOLUTION

3. We can now begin to understand what happened in our "SURPRISE" program:

   Line 20 draws a border around the high-resolution page.
   Line 40 switches to high-resolution page one without destroying our border.
   Line 50 switches to full page graphics.
   Line 60-70 draws a snowflake.

4. You may have discovered that by using the HGR command the High Resolution Screen is erased. How could we use POKE to our memory locations in order to switch from the Text Screen to the HGR Screen and back without erasing the HGR Screen?

   Acceptable answer:

```
10 REM TEXT TO HGR AND BACK
20 HGR
30 HCOLOR = 3
40 HPLOT 0,0 TO 279,159
50 HPLOT  279,0 TO 0,159
70 POKE -16303,O: REM TEXT
90 POKE -16304,0 : REM GRAPHICS
100 POKE -16297,0:  REM HI-RESOLUTION
110 END
```

Application:

1. How may we use our ability to control the graphic switches to write on both pages of the HIGH-RESOLUTION page and then flip back and forth?

   Study the following example:

```
10 HOME
20 HGR
30 HPLOT 140,100 TO 190,50 TO 90,50 TO 140,100
40 HGR2
50 HPLOT 140,0 TO 190,50 TO 90,50 TO 140,0
60 FOR I = 1 TO 5
70     POKE  - 16300,0: REM    SWITCH TO HGR1
80     GOSUB 150
90     POKE  - 16299,0: REM  SWITCH TO PAGE TWO
100    REM  FLIP PAGES ONCE SET
110    GOSUB 150
120 NEXT
130 TEXT : HOME
140 END
150 REM  STALLING LOOP
160 FOR A = 1 TO 1000: NEXT : RETURN
```

Summary:

1. How many graphics modes can the Apple II computer create?

2. How can we switch to a graphics page without erasing it?

3. How can we convert the HGR (page one) into a full graphics page? (After HGR, perform a POKE- 16302,0.)

4. How can we flip from HGR page one to HGR page two and back without erasing?

Assignmments:

1. Instead of using HGR, use only the switches shown in this lesson to draw a border around the HGR screen.

2. Write a program that draws a rectangle, and then switch to the text screen and print "THIS IS A RECTANGLE." Have this cycle continue three times without having to redraw the rectangle.

3. Modify the Surprise program so that an object of your own design is drawn on the screen without erasing the border.

Assignments for More Advanced Students:

1. Using the technique of flipping from HGR page one to HGR page two, create the illusion of motion.

2. Expand on the above challenge by drawing a new image on page two while showing an image on page one and then switching to page two. While page two is showing draw a new image on page one and so on, to create a more elaborate motion.

Topic 6
APPLE HI-RESOLUTION GRAPHICS — PART I

Performance Objectives: Students will be able to:

- plot points on the Apple II high resolution screen
- create figures and drawings on the high resolution screen
- use programs which switch from text to high resolution graphics of the Apple II computer
- understand and use the various graphic modes of the Apple II computer

Vocabulary: New: HGR, graphics page, HPLOT, HCOLOR
Review: text, stalling loop

Motivation:

- The computer's ability to display high resolution graphics is essential for many applications from computer assisted instruction (CAI) programs to simulation and game programs. Since computers differ in the way in which graphics are generated, this lesson will be machine specific to the Apple II, II+, and IIe computers with at least a 48K memory. Type the following program. Before you run it, can you predict what it will do?

```
10 HGR
20 HPLOT 0,0
30 HPLOT 279,0
40 HPLOT 279,159
50 HPLOT 0,159
60 HPLOT 140,80
70 END
```

(This program will place dots in the four corners of the HGR screen and one dot in the middle of the HGR screen.)

Aim: How can we use the high resolution mode of the Apple computer to create designs and shapes?

Development:

1. How can we program the computer to switch into the high resolution screen of the Apple computer? (Type HGR either in immediate mode or in a program.)

2. How can we switch from the high resolution screen back to normal text mode? (Type TEXT either in immediate mode or in a program.)

3. Imagine the HGR screen as a piece of graph paper. There are 280 columns (x positions from 0 to 279) and 160 rows (y positions from 0 to 159). In order to place a dot on any point of the HGR screen we must name an x and y position. The syntax to do this is HPLOT X,Y.

4. Try this:

```
10 REM draw a house
20 HGR: hcolor = 3
30 HPLOT 140,20 TO 180,40 TO 180,120 TO 100,120 TO 100,40 TO 140,20
```

5. In line 20 above the command HCOLOR = 3 tells the computer to draw in white. Normally, without this command, the computer will default to white anyway. However, if we wish to draw in another color we may choose from the list below:

color	number	color	number
black	0	black	4
green	1	orange	5
violet	2	blue	6
white	3	white	7

Try substituting different values for the HCOLOR statement and see what you get. (Note: You must have a color monitor for this.)

6. Discuss the various forms of the HPLOT command:

   a) HPLOT X,Y
   b) HPLOT X1,Y1 TO X2,Y2
   c) HPLOT X1,Y1 TO X2,Y2 TO X3,Y3 TO.....

7. Write a program that will draw a border around the HGR screen.

   Acceptable answer:

```
10 HGR
20 HPLOT 0,0 to  279,0 to 279,159 to 0,159
70 END
```

8. You may have discovered that by using the HGR command the high resolution screen is erased. How could we switch from the text screen to the HGR screen and back without erasing the HGR screen?

   Acceptable answer:

```
10 Rem text to hgr and back
20 HGR
30 HCOLOR = 3
40 HPLOT  0,0 TO 279, 159
50 HPLOT 279,0 TO 0,159
60 FOR I = 1 TO 3000: NEXT
70 TEXT
80 FOR I = 1 TO 3000: NEXT
90 POKE -16304,0 : REM graphics
100 POKE -16297,0:  REM hi res
110 END
```

(What is the purpose of lines 60 and 80? Stalling loops.)

Applications:

1. Modify the border program to draw a border around the HGR screen and then draw the two diagonals.

Acceptable answer:

```
10 HGR
20 HPLOT 0,0 TO 279,0 TO 279,159 TO 0,159
30 HPLOT 0,0 TO 279,159
40 HPLOT 279,0 TO 0,159
50 END
```

2. Take a piece of graph paper and mark 0 to 279 along the top and 0 to 159 along the left side. Then draw a desired shape by placing dots on the graph paper and connecting them with lines. At each point label the X and Y coordinates. Then write a short program to draw the shape using the HPLOT command discussed earlier.

Summary:

1. How do we program the Apple II computer to draw on the high resolution page? (HGR and HPLOTS)

2. What forms can we use for the HPLOT command?

3. How may we change colors on the high resolution page?

4. How may we switch from HGR to text without erasing what we have drawn on the high resolution page?

Assignments:

1. Using the HGR screen draw any 6-sided polygon.

2. Using the techniques discussed earlier, create a program that will show a house, then switch to the text screen which says "casa," then flip back to the house without redrawing it.

3. Create a program which will draw a happy face.

4. Create a program that will ask the student to select a happy or sad face. Then have the computer draw it.

5. Create a program that will draw a green tree and a blue house.

Assignments for More Advanced Students:

1. Write a program that will draw dots at random anywhere on the HGR screen.

2. Create a program that will draw the American flag.

Topic 7
APPLE HI-RESOLUTION GRAPHICS — PART II

Performance Objectives: Students will be able to:

- use programs which switch from high resolution graphics with text at bottom of page to full-page graphics
- learn to draw on the high resolution screen page 2 of an Apple II computer
- program the computer to flip graphics pages to simulate motion or for special effects
- understand the memory requirements of programming with graphics
- LOAD and SAVE APPLE graphics images

Vocabulary: New: HGR2, BLOAD, BSAVE
Review: HGR, graphics page, HPLOT, HCOLOR

Motivation:

- The computer's ability to display high resolution graphics is essential for many applications from computer assisted instruction (CAI) programs to simulation and game programs. Since computers differ in the way in which graphics are generated, this lesson will be machine-specific to the APPLE II, II+, and IIe computers with at least a 48K memory.

- We have all seen the technique of creating the illusion of motion by drawing images on each sheet of a pad of paper and then flipping the sheets. Take two 3 x 5 index cards. Draw an image on one card and the same image on another with only one small change. Tape the two cards together with the images facing out and a pencil in between with at least 3 inches extending from between the cards. Rub the pencil back and forth between the palms of your hands to produce a rapid flipping of the cards, thus creating an illusion of motion.

Aim: How can we use the high resolution pages of the Apple II computer to create designs and shapes and the illusion of motion?

Development:

1. In the previous lesson we discussed how the HGR command switches the Apple II computer into the high resolution screen. In addition, this mode provides for four lines of text at the bottom of the screen. This technique is used in many of the simulation programs which show a graphics screen and then at the bottom describe the scene and then inquire what action to take.

2. Can you modify the House Program of the previous lesson to ask a question at the bottom?

   Acceptable answer:

```
10 REM draw a house
20 HGR: HCOLOR = 3
30 HPLOT 140,20 TO 180,40 TO 180,120 TO 100,120 TO 100,40 TO 140,20
40 HOME
50 PRINT "What is the mathematical name for this shape?"
```

3. How can we switch from the high resolution screen with four lines of text at the bottom to a full page graphics page? After doing an HGR we can POKE -16302, 0.

4. When we are in full page graphics, we have converted the bottom text lines into part of our graphics page which now has 192 high-resolution horizontal lines available.

5. Write a program that will draw a border around the full HGR screen.

   Acceptable answer:

```
10 HGR: POKE -16302,0
20 HPLOT 0,0 TO 279,0 TO 279,191 TO 0,191
30 END
```

6. How can we switch from high resolution video back to normal text? (Type TEXT either in immediate mode or in a program.)

7. Now that we know there are two hi-res screens inside the Apple II computer (HGR, HGR2) let us examine where they are stored in memory. HGR is stored in memory location 8192 and goes to 16,383 and HGR2 is stored in location 16,384 to 24,575. At first these numbers can be intimidating, but they are simply addresses. Their equivalents in hexidecimal (a number system computers can understand based on powers of 16) are much easier to remember:

decimal	hexidecimal
8192	$2000
16384	$4000

   (Note: Hexidecimal numbers are preceded by a "$" to distinguish them from base 10 numbers.)

8. The actual hi-res page can be saved as a binary image on a disk and then loaded for future use. Thus, we can write a program that will draw a house and then save the image on disk. Another program could then load and display that image. How can we save and load hi-res images? After an image is drawn, such as we did earlier with the house, we can, in immediate mode, type the following:

   BSAVE program name, A$2000, L$2000

   If we wish to save the image on page two then we would type:

   BSAVE program name, A$4000, L$2000

   The image is stored as a binary image with the starting address being specified by the A parameter and the length is specified with the L parameter.

9. How can we retrieve the hi-res image saved on disk?

   Acceptable answer:

In immediate mode:

```
HGR
BLOAD program name
```

In a program:

```
100 HGR : D$ = CHR$(4)
110 PRINT D$; "BLOAD program name"
```

Applications:

1. How does real movement differ from simulated movement? Real movement involves a continuous physical motion which covers every point from start to finish. Simulated movement is the result of the computer's ability to draw, erase and redraw an image so quickly that it looks like real movement.

2. How can we flip between two hi-res screens, pages one and two, in order to create an illusion of motion?

   Run the following program:

```
10 HCOLOR = 3
20 HGR: HPLOT 279,0 TO 0,159
30 HGR2: HPLOT 0,0 TO 279,159
40 TEXT
50 POKE -16304,0: REM graphics
60 POKE -16297,0: REM Hi res
70 POKE -16300,0: REM page one
80 FOR I = 1 TO 2000: NEXT I
90 POKE -16299,0: REM page two
100 FOR I = 1 TO 2000: NEXT I
110 GOTO 70
```

Summary:

1. Why is it an advantage to have four lines of text at the bottom of a hi-res page? (for explanation, description and input)

2. How can we switch to full page graphics? (POKE -16302,0)

3. How can we switch from full page graphics back to text? (type TEXT)

4. How can we save the entire graphics page as a binary image?

   BSAVE program name, A$2000, L$2000 (page one)
   BSAVE program name, A$4000, L$2000 (page two)

5. How does real movement differ from simulated movement?

Assignments:

1. Write a program that will draw a boat on the hi-res page one.

2. Write a program that will draw a face on the hi-res page one. Then save

that image as a binary image on page one.

3. Write a program that will draw the same face as above except with one eye closed. Then save that image as a binary image on page two.

4. Using the flip-flop technique, BLOAD the hi-res faces created by assignment numbers 2 and 3 to see a face winking.

Assignments for More Advanced Students:

1. Write a program that will pick a number at random between 1 and 10 and asks the user to guess it. When the guess is correct, a happy face is drawn. When the guess is incorrect, a sad face is drawn.

2. Create 7 different hi-res images saving each on disk. Then create a slide show program which will show each image.

Topic 8
ANIMATION—PART I

Performance Objectives: Students will be able to:

- understand when moving graphics will add to the clarity of a program
- develop a list of situations that call for programs with moving graphics
- review simple techniques for moving block graphics
- review techniques for eliminating the possibility of getting an out of the screen's boundaries error
- learn a simple technique for preventing useful information from scrolling off the screen
- develop programs that can be used to scroll the text screen in any direction
- develop programs where all movement is controlled by the programmer and the user is a passive audience
- develop programs where movement can be controlled by the user

Vocabulary: New: cursor control, teaching program, motivational device
Review: scroll, concatenate, GET, INKEY$, CHR$

Motivation:

- If a picture is worth a thousand words then what is a moving picture worth? (Elicit student responses to arrive at the statement: Moving graphics are a great attention getter.)

- Is there a universal graphics mode that will work on any microcomputer? (One type of graphics that is supported by all microcomputers is the drawing of pictures on the text screen using whatever keys are present to write text.)

  - What is the advantage of using this type of graphics? (Make sure students understand that while this type of graphics is less accurate and colorful than low or high resolution graphics, the fact that it can be created on any machine makes it very useful to know. Simple BASIC statements such as PRINT, GOTO, GOSUB/RETURN, FOR/NEXT, IF/THEN, INPUT, INKEY$, GET, CHR$, and a thorough understanding of cursor control, are all that is needed to create useful moving graphics programs that are both fun to write and entertaining to view.)

Aim: What kinds of programs are enhanced by the introduction of moving graphics, and what kind of movement adds to a program's clarity?

Development:

1. Have students compile a list of situations where moving graphics will enhance a program.

   - Direct class discussion to an appreciation of capturing the attention of an audience in a teaching situation.

   - Develop an awareness of using moving graphics as a motivational device.

   - Remind students that historically men have communicated their thoughts through pictures since the days of the cave man. Master painters have

struggled to achieve depth and movement on their canvasses, and great sculptors strive to simulate movement in their statues. The need to create realism is therefore a universally accepted goal as is the need to communicate with others when faced with a language barrier. Thus, a picture comes to the rescue and one that moves adds both clarity and realism.

2. Once the need to immediately capture the attentionn of the user is established, ask the following question: When writing a teaching program, would you introduce your software with a still screen announcing your name, or would you prefer an introduction showing a school bus driving across the screen spelling out STUDENT SOFTWARE PRESENTS? Elicit the advantage of adding movement to the introduction. Plan the program.

   . Determine screen size.

   . Lay out bus on graph paper.

   - Experiment with keys to aid in diagram of bus. (O's for wheels,etc.)
   - Experiment with timing loop to create effective pause between letters. (not too long, not too short)
   - Determine screen line for title, determine top screen line for bus.
   - Review use of the semicolon with the PRINT statement.
   - Review GOSUB/RETURN procedures.
   - Review cursor control keys and clean screen procedures for your particular machine.
   - Review technique of using trailing spaces equal to the number of spaces that will create the movement. Explain to students that the longer the distance used to simulate movement, the faster the object appears to be moving. Since this is a school bus we want it to move slowly and will therefore advance only one space at a time.
   - The following is a suggested program for computers that support VTAB and HTAB.

```
5   REM BUS INTRODUCTION TO PROGRAM
10   H = 1
15   VTAB 10
20   HTAB H: PRINT "S"
25   GOSUB 200
30   HTAB H: PRINT "T"
35   GOSUB 200
40   HTAB H: PRINT "U"
45   GOSUB 200
50   HTAB H: PRINT "D"
55   GOSUB 200
60   HTAB H: PRINT "E"
65   GOSUB 200
70   HTAB H: PRINT "N"
75   GOSUB 200
80   HTAB H: PRINT "T"
85   GOSUB 200     (complete title)
90   END
100 REM THIS SECTION DRAWS BUS
200   H = H + 1
201   HTAB H: PRINT " XXXXXXXXXX"
204   HTAB H: PRINT " X SCHOOL '|"
```

```
205  HTAB H: PRINT " X    BUS : )"
206  HTAB H: PRINT "  |-O----O-/"
210  FOR X = 1 TO 1000: NEXT
215  VTAB 10
220  RETURN
```

3. Discuss advantages of allowing the user to control movement in a program. Introduce problem: Write a program that will permit the user to move a small arrow around the screen in any direction. Determine limits of screen area for moving the arrow.

   - Determine size of arrow.

   - Include spaces to erase arrow for each direction of its movement. Since arrow moves one space at a time it must be preceded by a space for a backwards move and followed by a space for a forward move. To move arrow up and down, the line above and below it must be packed with spaces. If the arrow occupies 3 spaces and 2 additional spaces are included for forward and backward movement, then how many spaces must we leave above and below the arrow? (4)

   - Show the value of positioning the cursor at the head of the arrow. Stress cursor control again. Remind students to leave space for the leading cursor when determining the screen limits.

   - Review concatenation. Explain the simplicity of using this type of string manipulation when creating moving blocks.

   - Review the ASCII Codes and the use of the CHR$ statement.

   - Review keyboard control when moving text diagrams. Discuss the value of selecting adjacent keys rather than U for up and D for down. Use simple IF/THEN statements to accomplish this procedure.

   - Stress that the selection of the lower limit of the screen should be high enough to disengage scroll. When we write on the bottom screen line the cursor will advance one line causing valuable material to scroll off the screen. Choosing our bottom line of spaces to fall on the next to last screen line will avoid this problem. Use simple programming statements such as IF V > 22 THEN V = 22.

   - A program that can be used for computers that support VTAB and HTAB is:

```
300  REM  SPACE ARROW
320  A$ = ">-<"
330  B$ = " "
335  C$ = "   "
340  V = 1:H = 1
350  VTAB V + 1: HTAB H: PRINT B$ + C$: GOTO 370
360  VTAB V - 1: HTAB H: PRINT B$ + C$
370  VTAB V: HTAB H
380  PRINT B$ + A$ + B$
385  IF V = 23 THEN V = V - 1: VTAB 23: HTAB 1: PRINT "     "
390  VTAB V: HTAB H + 4
400  GET G$
410  IF G$ =  CHR$ (73) THEN V = V - 1: GOSUB 460: GOTO 350
```

```
420  IF G$ =  CHR$ (74) THEN H = H - 1: GOSUB 460: GOTO 370
430  IF G$ =  CHR$ (77) THEN V = V + 1: GOSUB 460: GOTO 360
440  IF G$ =  CHR$ (75) THEN H = H + 1: GOSUB 460: GOTO 370
450  IF G$ <> CHR$(73) OR G$ <> CHR$(74) OR G$ <> CHR$(75) OR
       G$ <> CHR$(77) THEN 400
460  IF H < 1 THEN H = 1
470  IF H > 35 THEN H = 35
480  IF V < 1 THEN V = 1
490  IF V > 22 THEN V =  22
500  RETURN
```

4. Now that we have learned how to disengage a scroll, let's see if we can make the screen scroll in a different direction. Suppose we wish to clear our screen with a right scroll, what are some of the things we have to consider before we write the program?

   - A vertical boundary should be drawn which appears to erase the screen. This boundary can be created with a simple FOR/NEXT loop.

   - The screen can be erased in a vertical manner by filling each vertical line with blank spaces. After each line of spaces is drawn the cursor should be returned to the topmost screen position in preparation for drawing the next line of spaces.

   - The boundary must be erased and redrawn each time the cursor returns to the top of the screen so that movement can be simulated.

   - To erase the final column, the cursor must be disengaged to prevent it from advancing to the next line. The use of the semicolon with the PRINT statement is useful in this instance.

   - Review nested FOR/NEXT loops. Elicit their usefulness for this program.

   - A program that can be used with computers that support VTAB and HTAB is:

```
80   REM   RIGHT SCROLL
90   REM DRAW BOUNDARY
100  VTAB 1
110  FOR X = 1 TO 23
120  PRINT "!"
130  NEXT X
135  REM ERASE AND REDRAW BOUNDARY
140  FOR Y = 1 TO 39
150     FOR X = 1 TO 23
160        VTAB X: HTAB Y: PRINT " !"
170     NEXT X
180  NEXT Y
```

   (Note: This program does not begin by clearing the screen. Running the program while the listing is still on the screen will demonstrate the advantages of this type of a scroll.)

5. Remind students of their experiences with overlapping triangles early in their geometry course. Elicit the different techniques that were used to

enhance their understanding of this rather confusing topic. Ask students to suggest programs that aid in the clarification of this topic. Assist students in writing a program where the user can separate the triangles by simply pressing a key. Plan the program.

- Select the areas of the screen to be used for diagrams and those used for instructions.

- Select the keys that will create the most realistic triangle.

- Determine how to position the triangles in their original position and in their separated condition.

- Use graph paper to lay out the triangles.

- Discuss the use of concatenation and simple PRINT statements. Determine that this situation is best served with PRINT statements.

- Compare the GET, INPUT, AND INKEY$ statements. Extol the user-friendly virtues of "Press any key to..." and stress the necessity of selecting a specific key to terminate a program.

- Remind students to label the vertices of the triangles and to provide for a return to the original diagram.

- The following is a suggested program for computers that support HOME and GET.

```
100  REM OVERLAPPING TRIANGLES
110  REM DRAW ORIGINAL TRIANGLES
120  HOME
130  PRINT "A        B"
140  PRINT "B|      /B"
150  PRINT "B |    / B"
160  PRINT "B  | /  B"
170  PRINT "B   X   B"
180  PRINT "B  / |  B"
190  PRINT "B /    | B"
200  PRINT "B/      |B"
210  PRINT "---------"
220  PRINT "E        F"
225  REM SKIP LINES AND PROVIDE FOR USER CONTROL TO SEPARATE
     TRIANGLES
230  PRINT : PRINT : PRINT "PRESS ANY KEY TO SEPARATE TRIANGLES"
240  GET G$
241  REM CLEAR SCREEN AND DRAW SEPARATED TRIANGLES
242  HOME
245  PRINT "A                    B"
250  PRINT "B|                  /B"
260  PRINT "B |                / B"
270  PRINT "B  |              /  B"
280  PRINT "B   |            /   B"
290  PRINT "B    |          /    B"
300  PRINT "B     |        /     B"
310  PRINT "B      |      /      B"
350  PRINT "---------    ---------"
```

```
360  PRINT "E        F    E        F"
365  REM SKIP LINES AND PROVIDE FOR USER CONTROL
370  PRINT : PRINT : PRINT
380  PRINT "PRESS ANY KEY TO RETURN TO ORIGINAL"
385  PRINT "DIAGRAM"
386  PRINT : PRINT : PRINT "PRESS E TO END DEMONSTRATION"
390  GET G$
395  REM PERMIT USER TO END PROGRAM OR RETURN TO ORIGINAL DIAGRAM
400  IF G$ = "E" THEN  END
410  GOTO 120
```

Applications:

1. Write a program that will erase the contents of the text screen with a left scroll.

2. Change the school bus program to simulate "Sky Writing" by an airplane.

3. Modify the space arrow program to print a left arrow that moves up or down in the last eight screen columns. Use the ")" key to create a screen boundary. Suggested arrow, "<--<<". We wish to keep our arrow to the left of the screen boundary.

Summary:

1. Why are moving graphics such an excellent motivational tool?

2. Can you name a situation where movement would enhance the graphics?

3. Are there situations where movement would serve to clarify the program's message?

4. What are the commands that we use to position the cursor?

5. What techniques are used to erase text from specific areas of the text screen?

6. Why is it necessary to know how to scroll the text screen in different directions?

7. How can we disable a SCROLL UP?

8. What advantage is there in allowing the user to control the movement of a program?

Assignments:

1. Write a program that will erase the text screen with a SCROLL DOWN.

2. Draw a parallelogram with one diagonal. Label the vertices. Separate the triangles formed by the diagonal, preserving the labels. Turn one of the triangles around to stress congruence of corresponding parts. Provide for a return to the original diagram as well as an exit from the program.

Assignment for More Advanced Students:

1. After completing assignment 2, program moving arrows to highlight corresponding parts. Use appropriate pauses and messages to explain the correspondence of the parts.

Topic 9
ANIMATION—PART II

Performance Objectives: Students will be able to:

- appreciate the way moving graphics add clarity to demonstration programs
- develop demonstration programs where movement can be controlled by the programmer
- use the PRINT statement in conjunction with a pause to draw and then erase objects on the text screen
- draw and erase objects from a text screen without deleting valuable existing text
- understand the need for careful planning of screen formatting before any program coding can begin
- develop a technique which will enable objects to appear to move in different directions at the same time

Vocabulary: New: demonstration program, optical illusion
Review: scroll, screen input area, screen output area

Motivation:

- In our first lesson, we saw how moving graphics could be used to enhance a teaching program. Are there other types of programs that would benefit from the introduction of moving graphics? (demonstration programs)

- Animation has always been a useful tool for the advertiser. Can you think of any situations where a manufacturer could demonstrate the usefulness of a product by a moving graphics program?

Aim: How can moving graphics be used to enhance a demonstration program?

Development:

1. We have all been alerted to the values of a healthy diet. There has been much discussion and literature concerning the reduction of cholesterol levels in the blood. A manufacturer of cooking oil wishes to alert the public to the values of purchasing his product. He develops the following diagram:

CROSS SECTION OF ARTERIES:

```
   AAAAAAA            AAAAAAA
  AOOOOOOOA          AOOOOOOOA
 AO       OA        AO       OA
AO         OA      AO OOO  OOOOA
AO         OA      AOOOOOO OOOOA
AO         OA      AOOOOOOOOOOOA
 AO       OA        AOOOOOOOOOA
  AOOOOOOOA          AOOOOOOOA
   AAAAAAA            AAAAAAA
```

NORMAL ARTERY (wide opening and smooth inner lining)	DISEASED ARTERY (narrow opening and inner lining caked with deposits)

THREE FAT MOLECULES

```
 H H H H H H H      H H H H   H H      H   H   H   H
 : : : : : : :      : : : :   : :      :   :   :   :
:C:C:C:C:C:C:C:    :C:C:C:C:C:C:C:    :C:C:C:C:C:C:C:
 : : : : : : :      : : :   : : :        :   :   : :
 H H H H H H H      H H H   H H H        H   H   H H
```

SATURATED FAT MONOUNSATURATED FAT POLYUNSATURATED FAT

- In the diagram he not only shows the structure of different fat molecules, but he offers a representation of a healthy and a diseased artery. What are some of the ways that we can add motion to this diagram to aid the manufacturer in getting his message across to the public?
- Begin with a saturated fat molecule and slowly erase the hydrogen atoms until the polyunsaturated molecule emerges.
- Reverse the procedure and end with the additional hydrogen molecules flashing on the screen.
- Begin with a healthy artery and slowly build up the caked residue on the arterial walls, depicting the diseased artery.
- Reverse the procedure.
- Show the smooth flow of blood through a healthy artery as opposed to a more sluggish interrupted flow, through the diseased artery.
- Direct student discussion to establish that this demonstration could be used in a classroom as part of a hygiene lesson.

2. Demonstration programs are valuable aids in all areas. The science classroom is no exception. Consider the following diagram of a battery, used to explain Ohm's law.

THE OHM'S LAW
A RULE OF RESISTANCE

0.5 AMP current

```
          >>--->
        ____________________
   ^   /                    |
   !   !      <---<<        !
   ^   !                    #
   ^   !     ________      !^  ##        ______
       !     !  zinc      !!  ####--/        |
     c +++   --- plate     !^  ####====-)      !
  X  o +++   ---    X      !^  #filament)      !
  X  p +++   ---    X      !   ####====-)      !
  X  p +++   ---    X      !   ####--|______/
  X  e +++   ---    X      !   ##
  X  r +++   ---    X      !<-<#
  X    +++   ---    X      !___!
  X    +++   ---    X
  X    +++   ---    X
  X    +++   ---    X
  X    +++   ---    X
  X    +++   ---    X
  X      <--<<      X
  X                 X
  X   conducting    X
  X  fluid (brine)  X
  XXXXXXXXXXXXXXXXXX
```

1.5 VOLT BATTERY

- Elicit that the movement of a stream of electricity would be very effective in making this diagram come to life. This diagram is formatted for a 40-column screen. A discussion of preplanning screen positions would be very useful here. If it does not fit your particular screen it can be easily altered, or only a segment of the screen need be used.

- Offer two types of movement:

  - Arrows that move to show the direction of the current.

  - The entire current moves, through the solution, through the filament, etc.

- Elicit that a program showing the movement of an electrical current can be easily accomplished. It would not only lend realism to the diagram but it would aid in holding the viewer's attention.

- Plan the program:

  - Establish the screen positions for the path of the current.

- Select the appropriate keys to aid in creating the illusion of a moving current. Right ">", left "<", up "^", down "V".

- Experiment with length of timing loops.

- Establish cursor positions and discuss cursor control keys.

- Review the use of the semicolon with the PRINT statement.

- Review the use of GOSUB/RETURN. Discuss the value of keeping the timing loop in a subroutine.

- Experiment with INVERSE and FLASH as opposed to normal mode. Determine that FLASH detracts from the demonstration whereas INVERSE enhances the electrical stream.

- This sample program will work with a computer that supports VTAB and HTAB.

```
10  REM  ELECTRICAL PATH DEMO
20  HOME
30  INVERSE
40  V = 9
50  FOR X = 1 TO 8
60      VTAB V
70      PRINT "^"
80      V = V - 1
90      GOSUB 640
100 NEXT X
110 VTAB 1
120 FOR X = 1 TO 20
130     PRINT ">";
140     GOSUB 640
150 NEXT X
160 FOR X = 1 TO 7
170     HTAB 20: PRINT "V"
180     GOSUB 640
190 NEXT X
200 HTAB 20
210 FOR X = 1 TO 5
220     PRINT "<";
230     GOSUB 640
240 NEXT X
250 FOR X = 1 TO 5
260     HTAB 25: PRINT "V"
270     GOSUB 640
280 NEXT X
290 H = 25
300 FOR X = 1 TO 5
310     HTAB H: PRINT "<";
320     H = H - 1
330     GOSUB 640
340 NEXT X
350 FOR X = 1 TO 5
```

```
360      HTAB 20: PRINT "V"
370      GOSUB 640
380  NEXT X
390  H = 20
400  FOR X = 1 TO 3
410      HTAB H: PRINT "<";
420      H = H - 1
430      GOSUB 640
440  NEXT X
450  V = 18
460  FOR X = 1 TO 14
470      VTAB V
480      HTAB 17: PRINT "^"
490      V = V - 1
500      GOSUB 640
510  NEXT X
520  VTAB 5:H = 16
530  FOR X = 1 TO 8
540      HTAB H: PRINT " ";
550      H = H - 1
560      GOSUB 640
570  NEXT X
580  FOR X = 1 TO 5
590     HTAB 8: PRINT "V"
600     GOSUB 640
610  NEXT X
620  NORMAL
630  END
640  FOR T = 1 TO 50: NEXT T
650  RETURN
```

3. Another useful demonstration program is one that can be used in the biology classroom. The division of cells by mitosis is depicted by the following diagram.

REPRODUCTION BY MITOSIS:

Most cells reproduce by MITOSIS, the process in which generic information is transmitted to each offspring cell.

```
      CCC                CCC                  CCCC
    CC   CC            CC  CC                C    C
   CC     CC          CC (*) CC             C      C
  CC       CC        CC       CC           C  (*)   C
  CC   (*)  CC       CC    (*) CC          C (%)    C
  CC        CC       CC        CC           C      C
  CC        CC       CC        CC           C      C
  CC   (%) CC        CC  (%)  CC             C     C
 CC        CC       CC (%)    CC           CC (%)   C
 CC       CC        CC       CC           C    (*)   C
 CC      CC         CC      CC            C          C
  CCCCCCC            CCCCCCC               CCCCCCCCCC

 CELL WITH TWO       CHROMOSOME            PREPARATION FOR SPLITTING
 CHROMOSOMES         REPRODUCTION
```

```
      CCC                 CCC
    CC   CC             CC  CC
   CC     CC           CC (*) CC
  CC       CC        CC       CC
 CC   (*)  CC        CC       CC
 CC         CC      CC        CC
  CC         CC    CC         CC
  CC   (%)  CC      CC          CC
 CC         CC       CC (%)     CC
 CC        CC        CC        CC
 CC       CC         CC       CC
  CCCCCCC             CCCCCCC
```

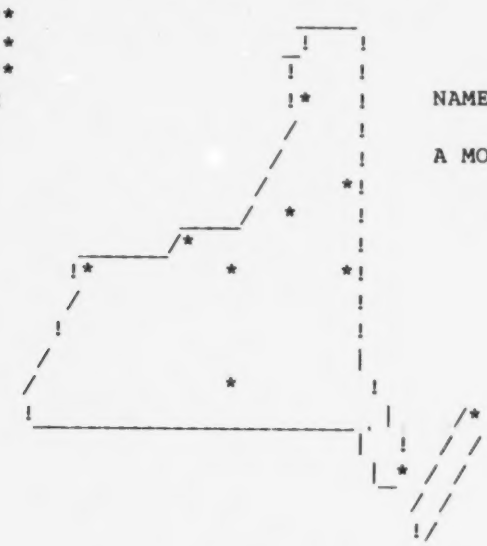
TWO NEW IDENTICAL OFFSPRING CELLS

- Remind students of the program they wrote in the previous lesson which separated triangles.

- Lead a class discussion in the structure of this program but leave time for a class exercise.

4. A social studies class might benefit from a demonstration map lesson. We could point out cities in New York State by an arrow that we draw and then erase from the screen. Offer the class the following map:

```
*******
* 100 *
*******
 SCORE
```



NAME THE CITIES IN NEW YORK STATE

A MOVING ARROW WILL POINT TO A CITY

IF YOU NAME THE CITY CORRECTLY THE ARROW WILL JUMP TO ANOTHER CITY

IF THE ARROW REMAINS IN THE SAME LOCATION THEN YOUR ANSWER IS INCORRECT

EACH CORRECT RESPONSE WILL RECEIVE 10 CREDITS

3 CREDITS WILL BE DEDUCTED FOR EACH INCORRECT RESPONSE

YOUR SCORE WILL BE TALLIED IN THE UPPER LEFT HAND CORNER OF THE SCREEN

*** GOOD LUCK! *** *** PRESS RETURN TO BEGIN. ***

- The directions under the map indicate one way to write the program. Discuss other ways that this program can be presented to students. In all cases provide for some user input, either by writing the full name of the city, or simply a letter that will represent the name of the city. Stress the need for establishing an area on the screen which is used solely for user input.

- Plan the program:
  - When drawing the map, make sure that enough space is left around each city, to draw and erase an arrow without deleting any text. If the arrow is to occupy three spaces then there must be three blank spaces left either to the right or left of the star. Provide for disabling the scroll. We do not wish to lose the top of our map.

  - Make timing loop long enough for user to have enough time to think of the correct answer.

  - A flashing arrow might be a good idea to alert the user to direct his attention to the correct screen area.

  - This program lends itself to subroutines. One subroutine could be set up to draw and erase the arrow at various screen positions. Another subroutine could position the cursor at the beginning of the input box, awaiting the user's input. The cursor could then be returned to that exact position awaiting the next input. By forcing the cursor back to a given position we are in effect turning off the scroll.

  - A simple way of doing this for a machine that supports VTAB and HTAB is in the following program segment. (No attempt is made here to relate this arrow to the above map. This is offered only as a means of drawing and erasing arrows at different screen positions.)

```
495  HTAB 1: VTAB 2
500  INPUT A$
501  IF A$ = "A" THEN H = 5:V = 8
502  IF A$ = "B" THEN H = 10:V = 15
510  IF A$ = "C" THEN H = 12:V = 10
511  IF A$ = "D" THEN H = 20:V = 3
512  IF A$ = "E" THEN H = 3:V = 7
520  GOSUB 910
600  GOTO 495
910  HTAB H: VTAB V: PRINT     "}}===}"
920  FOR X = 1 TO 2000: NEXT
930  HTAB H: VTAB V: PRINT "       "
940  RETURN
```

  - Point to the score box at the top of the map. Discuss with the class a subroutine that can be written to keep score. This can be left for a class exercise.

5. Remind students of the classic "upstream/downstream" problems that they encountered in their algebra classes. Discuss the type of optical illusion that must be created if we wish to see a boat move with the current and against the current on the same screen. Plan a program that could be used for this demonstration.

- Create the river with an appropriate key. The "^" key can be used to simulate waves.

- Remind students that the boat moves faster with the current (downstream), and slower against the current (upstream).

- Divide the stream in half, one side for the upstream demonstration and the other side for the downstream.

- Create two strings, one for the boat that will move upstream and the other for the boat that will move downstream.

- Instead of using spaces to erase this time we will erase with the waves. Thus, we need a string that will erase the boats.

- Remind students that the greater the distance covered the faster the object appears to move. Hence, the upstream boat will be redrawn at closer screen intervals then the intervals we use for the downstream boat. Agree to advance the upstream boat one line and the downstream boat three lines.

- Add to the illusion by starting the downstream boat at the top of the screen and the upstream boat from the bottom of the screen.

- Provide for the return of the faster boat to the top of the screen after it reaches the bottom screen position. This will turn off the scroll and emphasize the difference in rates.

- Position final cursor under river at end of program. This will prevent the cursor from destroying the screen display.

- A program that can be used with computers that support VTAB and HTAB follows.

```
5 REM UPSTREAM/DOWNSTREAM DEMO
10  A = 1:B = 22
20  B$ = "^^^^^^^^^^^^"
30  C$ = "^^|=/ ^^^^^^"
35  D$ = "^^^^^^|=/ ^^"
40  HOME
45  REM DRAW STREAM
50  FOR T = 1 TO 22
60  PRINT B$
70  NEXT T
75  REM DRAW DOWNSTREAM BOAT
80  VTAB A: PRINT C$
85  REM DRAW UPSTREAM BOAT
90  VTAB B: PRINT D$
100  GOSUB 1000
105  REM ERASE BOATS TO CREATE MOVEMENT
110  VTAB A: PRINT B$
115  VTAB B: PRINT B$
120  REM PROVIDE FOR INCREMENTS AND SCREEN LIMITS
121  A = A + 3
122  IF A > 22 THEN A = 1
125  B = B - 1
```

```
126  IF B = 1 THEN 1030
127  REM REDRAW BOATS
130  GOTO 80
140  REM ERASE BOATS AND REDRAW WAVES
1000  FOR X = 1 TO 200: NEXT X
1020  RETURN
1025  REM POSITION CURSOR UNDER RIVER TO PREVENT INTERFERENCE WITH
      SCREEN DISPLAY
1030  VTAB 23: GET G$
1040  END
```

Applications:

1. Write a program that will draw a simple circuit. Provide for the closing of a switch and depict the path of the current. Open the switch and stop the flow of the current.

2. Draw a floor plan of the main floor of your school. Draw and erase a flashing arrow that will aid in depicting strategic locations. Use arrows moving up and down to show staircase routes. Save this plan to be used as a subroutine later as part of a freshman orientation program.

3. Develop a program that will show the evaporation of moisture from the ocean, the formation of clouds and the water returning to earth in the form of rain.

Summary:

1. How can moving graphics enhance a demonstration program?

2. Are blank spaces always used to erase objects from a text screen?

3. What is the purpose of providing for an input box?

4. What technique is used to show objects moving at different speeds?

5. Is it possible to create an illusion of objects moving in different directions on the same text screen?

6. What problems can occur if we do not format our screen positions in advance of writing our programs?

7. When is it best to define our graphics as a string and when would it be best to use simple print statements?

Assignments:

1. Write a program that can be used in a biology class to demonstrate reproduction by mitosis. Provide for sections that show the doubling of the chromosomes and the splitting of the cell.

2. Write a subroutine that can be used to keep score in the map program. Make sure that the cursor is returned to the score box before each score is printed.

3. Write a program that demonstrates the path of an electrical current in a parallel circuit. Provide for the opening and closing of both switches. Show the action of the current in all four positions.

Assignment for More Advanced Students:

1. Write a program that will show the action of the wind on a plane. Use the same screen to depict the plane flying against the wind and with the wind (refer to the upstream/downstream program we developed in class).

Topic 10
ANIMATION TO ENHANCE CAI—PART I

Performance Objectives: Students will be able to:

- write programs that draw cartoon characters on the text screen
- create programs that will enable these characters to move
- establish when a program will benefit from the addition of a cartoon character
- select specific drawings to enliven different types of programs
- adapt previously programmed cartoons to convey a different message
- review techniques for creating various screen scrolls and enliven their appearance with the addition of a cartoon character
- establish criteria for selection of a cartoon type

Vocabulary: New: simulation, appropriate cartoon selection
Review: teaching program, scroll, pause

Motivation:

- When writing a program to help someone review basic skills, it becomes necessary to inform the user if he has answered a question correctly, or if he has to go back and try again. What are some of the techniques that we can use to make this part of the program pleasing to the user? (Introduce a cartoon that conveys the appropriate message.)

- Are there any universal standards that we should be aware of before designing our cartoon? (A correct answer would call for a smile, a star, etc. Wrong answers should be rejected with words of encouragement, e.g., Hope that success is just around the corner.)

- Are there any other considerations before we draw our cartoon? (The age of the user, the subject of the program, the size of the screen, etc.)

Aim: How can moving cartoon characters be an effective part of the teaching programs that we write?

Development:

1. Would it be wise to greet a first grader who spelled the word cat, "Kat", with a violent explosion accompanied by loud bangs, followed by a flashing monster getting ready to gobble him up because of his mistake? Establish a list of criteria that students can refer to before planning their cartoon (see below).

   - Size of screen or specific screen area where cartoon will appear.

   - Can the cartoon be related to the subject? This is an excellent opportunity for students to let their imaginations soar. If the responses are feeble, then the teacher can suggest some of the following:

     a) walking test tubes
     b) dancing numbers or letters
     c) paintbrush writing words

d) single-cell creatures
e) figures with computers for heads
f) basketballs passing through hoops
g) clothing that suits a historical period (a three-cornered hat)

- What is the age of the user? Small animals are good for younger students while teenagers prefer aliens and monsters. The cartoon should be adapted to convey different messages without laborious additional programming.

- Does the action taken by the cartoon help to convey the message, or will it detract from what we are trying to say?

- Is the time taken to show the cartoon so long that the user gets bored waiting for the routine to end?

- Is the time too short, leaving the user with a feeling that he missed something?

2. Develop a cartoon character to tell the user he made an error.

- Draw a picture of an alien.

- Use the "X" key and the minus sign to draw the alien.

- Put a frown on his face.

- Have the alien tell the user that he is wrong, but give him a chance to try again.

- Use sound effects, a bell or a buzz.

- Make a picture that you can use with all school grades.

- Suggested cartoon:

```
  X     X
   X   X
    X X
   WRONG
  X     X
 X  | /  X
 X  o o  X
 X   :   X
 X  /-|  X
  X     X
   -----
  X TRY X
 XX  A  XX
 XX  G  XX
 XX  A  XX
XXX  I  XXX
  X  N  X
  XWRONGX
   XX XX
   XX XX
   XX XX
  XXX XXX
```

68

3. Adapt the above cartoon to convey a message of success.

- Change the frown to a smile.
- Change the "X" to a "*"
- Change the "-" to a "+"
- Change the message.
- Suggested cartoon:

```
 *     *
  *   *
   * *
  RIGHT
 *     *
*  ° °  *
*  o o  *
*   :   *
*  |_/  *
 *     *
  +++++
 * YOU *
**     **
**  A  **
**  R  **
***  E  ***
 *     *
 *RIGHT*
  ** **
  ** **
  ** **
 *** ***
```

4. Now we are ready to breathe some life into our characters. Establish with students the necessity for different movements.

- Develop a list of suggested movements for Mr. Right.
  - He could jump up and down in one spot on the screen.
  - He could jump across the screen, at the same time clearing it of any text
  - He could be shown skipping rope.
  - He could be accompanied by some happy sounds (bells, clapping) and appear randomly in several different screen positions. (For example, position him at 3 random locations for 5 or 10 seconds each.)
  - He could accompany a right scroll across the screen.

. Suggested movements for Mr. Wrong.

- His time on the screen should be shorter than the time that Mr. Right appears. (We assume the student will be anxious to try and find his error, rather than sit back and be pleased with his success.)

- We can try to program him to do the opposite of what Mr. Right did:

  a) skip rope, but miss
  b) emerge feet first from the top of the screen to the bottom accompanied by the sound of a bomb dropping
  c) accompany a left scroll across the screen
  d) make him appear at only two random locations

- He could stand still and have his antennae drop.

5. Make use of the natural scroll up of the screen to change pages with a little pizazz. Write a program that shows a cartoon character pushing the screen text up and out of sight. Have the user control the motion by pressing a key when he is ready to see the next screen. Assume that you are writing a spelling practice program. Write a very simple introduction to the program and then include your scroll.

. Plan the program.

- Review the string manipulation statements (LEN(),MID()).

- Review PRINT TAB(), and the method of incrementing each tab stop.

- Plan to print "spelling practice," one character to a line on a diagonal line, centered on the text screen. Use a pause to spell out each letter separately.

- Print a message at the bottom of the screen in inverse letters instructing the user to "press return to continue."

- Review the CHR$() statement and the ASCII codes.

- Create a boundary with a FOR/NEXT loop.

- Create a cartoon character that appears to be pushing the text screen up.

- Use a short pause after each line of the cartoon is drawn to slow the action. (This is not included in sample program in an attempt to save space. Program the pause every other step of the cartoon or use GOSUB/RETURN.)

- Allow user control for continuation of the next segment of the program.

Sample program:

```
10  REM  INTRODUCTION
15  REM Clear the screen
20  S$ = "SPELLING PRACTICE"
```


BEST COPY AVAILABLE

```
30  FOR I = 1 TO  LEN (S$)
40     PRINT  TAB( I + 10) MID$ (S$,I,1)
45     FOR X = 1 TO 500: NEXT
50  NEXT I
60  PRINT : PRINT : PRINT : PRINT
61  REM HALT PROGRAM UNTIL USER IS READY TO CONTINUE
70  INPUT "PRESS RETURN TO CONTINUE";A$
80  IF A$ =  CHR$ (13) THEN 85
82  REM DRAW BOUNDARY
85  FOR X = 1 TO 40
90     PRINT "=";
100  NEXT X
105  REM DRAW CARTOON
110  PRINT "            W    W   W    W"
120  PRINT "            V    !   !    V"
130  PRINT "            V    °°°°°    V"
140  PRINT "            V    OOOOO    V"
150  PRINT "            V   ^      ¨  V"
160  PRINT "            V   ^ ¿ ¿  ¨  V"
170  PRINT "            V   ^  °   ¨  V"
180  PRINT "            V ^^^  V  ····· V"
190  PRINT "            V    ^    ¨    V"
200  PRINT "             V    ^X¨    V "
210  PRINT "              V  ^^X···  V"
220  PRINT "               V^^^X·····V"
230  PRINT "               ^^^^X·····"
240  PRINT "               VVVVVVVVV"
250  PRINT "                VVVVVVV"
260  PRINT "                 VVVVV"
270  PRINT "                  VVV"
280  PRINT "               /   L L"
320  PRINT : PRINT : PRINT : PRINT
322  REM RETURN PROGRAM CONTROL TO USER
330  INPUT "PRESS RETURN TO CONTINUE";A$
350  IF A$ =  CHR$ (13) THEN 400
400  REM Clear the screen and complete the program
```

Applications:

1. Write a program that will cause a rocket to fly horizontally across the top of the text screen in response to user input. If the answer is correct, fly the rocket from left to right. Make sure the nose of the rocket is pointing in the direction of flight and trail an appropriate success message. If the answer is wrong, draw the rocket in the opposite direction, have it fly across the screen from right to left and trail some words of encouragement. Cover greater screen distances with the "Wrong" rocket so it appears to move faster then the "Right" rocket.

2. Modify the Right Scroll program we developed at our previous lesson to include a cartoon character. Have the character appear to push the boundary across the screen. Provide for user control over the program (i.e. let the user decide when to turn the page). Print an appropriate message at the bottom of the screen instructing the user what key to press. Force the user to press the correct key by not accepting a key other than the one specified.

3. Write a program that will print five true/false questions. The questions are to occupy the left three fifths of the text screen. Divide the remaining two-fifths of the screen in half. Use columns 24 through 31 for the input area where answers will be written. Position the cursor at the end of Question 1. If the correct answer is given, then have the cursor jump to the end of Question 2. If the incorrect answer is given, then return the cursor to its original position at the end of Question 1. In this manner, provide for the answering of all five questions. Use the last eight columns to draw a vertical rocket. Program the rocket to land at the bottom of the screen. Every time a correct response is given, advance the rocket several screen lines downward. Do not move the rocket for an incorrect answer, but you may ring a bell if you wish. After all five questions have been answered correctly, the rocket should rest on the landing pad and the program should end.

. Suggested program for computers that support VTAB and HTAB:

```
5   REM TRUE FALSE TEST
7   V = 1: HOME
9   REM THIS SECTION PRINTS QUESTIONS
10   PRINT "ANSWER TRUE OR FALSE"
20   VTAB 4: PRINT "QUESTION 1"
30   VTAB 8: PRINT "QUESTION 2"
40   VTAB 12: PRINT "QUESTION 3"
50   VTAB 16: PRINT "QUESTION 4"
60   VTAB 20: PRINT "QUESTION 5"
62   REM LANDING PAD FOR ROCKET
65   VTAB 23: HTAB 30: PRINT "~~~~~~~"
67   REM THIS SECTION FOR USER RESPONSES
70   VTAB 4: HTAB 24: INPUT A$
80   IF A$ = "TRUE" THEN 100
90   GOTO 70
100   GOSUB 1000
110   VTAB 8: HTAB 24: INPUT B$
120   IF B$ = "FALSE" THEN 140
130   GOTO 110
140   GOSUB 1000
150   VTAB 12: HTAB 24: INPUT C$
160   IF C$ = "FALSE" THEN 180
170   GOTO 150
180   GOSUB 1000
200   VTAB 16: HTAB 24: INPUT D$
210   IF D$ = "TRUE" THEN 230
220   GOTO 200
230   GOSUB 900
250   VTAB 20: HTAB 24: INPUT E$
260   IF E$ = "TRUE" THEN 280
270   GOTO 250
280   GOSUB 900
300   END
900   REM THIS SECTION PRINTS AND MOVES ROCKET
1002   FOR A = 1 TO 3
1003       VTAB V: HTAB 32: PRINT "      "
1005       HTAB 32: PRINT "  ß  "
1010       HTAB 32: PRINT " / | "
1020       HTAB 32: PRINT " ßUß"
```

```
1030     HTAB 32: PRINT " ßSß"
1040     HTAB 32: PRINT " ßAß"
1050     HTAB 32: PRINT " | / "
1060     HTAB 32: PRINT "  ^"
1070     V = V + 1
1080  NEXT A
1090  RETURN
```

Summary:

1. What are some considerations that we must keep in mind when selecting cartoons for our teaching programs?

2. Is it wise to create cartoon characters that occupy the entire screen?

3. Why is it necessary to consider the age of the user when designing cartoon characters?

4. What is the purpose of using specific screen areas for user input?

5. When creating moving cartoon characters, how can we insure that valuable text will not be deleted?

6. What is the benefit of letting the user control the turning of the screen page?

7. What are some techniques we use to insure that the user does not get bored with repetitive cartoons?

8. Is it possible to suit a cartoon to a particular program?

Assignments:

1. Create a cartoon character that can be used to introduce tenth grade students to the BASIC language. Plan a lesson that will teach students the difference between PRINT X and PRINT "X". Depict the results of utilizing at least three PRINT instructions.

2. Write a subroutine that prints a cartoon character on the bottom half of the screen. Use the right and left arrow keys to move the cartoon back and forth. Make sure that no out of bounds error occurs.

3. Write a subroutine that draws a dog with a long tail and pointy ears. Wag the tail and print a success message. Drop the ears and the tail and print a message that encourages the user to go back and try again.

Assignment for More Advanced Students:

1. Write a program that spells five words incorrectly. Every time the user corrects one of the words, replace the incorrect word with the correct one and add a flashing star. Introduce a cartoon character into the program that comments on the progress of the user. Program the cartoon to move in a specific screen area.

Topic 11
ANIMATION TO ENHANCE CAI—PART II

Performance Objectives: Students will be able to:

- program the computer to keep track of a user's score
- develop interesting ways of displaying that score
- design figures in various positions in an effort to enhance movement
- learn a sequence for drawing and erasing these figures that will make movement more realistic
- design programs that provide remedial practice for users, that are written on the text screen, and that include moving cartoon characters

Vocabulary: New: realistic movement
Review: simulation, counter, accumulator

Motivation:

- What is the purpose of keeping score in a program? (To make the user aware of her proficiency. To challenge user to improve her skill.)

- What are some of the ways that a computer can keep track of a user's score? (Set a variable to be a counter. Store the results of a user's attempts in an accumulator.)

- What are some of the ways that a computer can display this score? (A numerical display in a specific area of the screen. Some type of graphic display.)

- How can cartoon characters enhance the process of keeping score? (An interesting display provides more of a challenge for the user.)

- Can moving cartoon characters add another dimension to the procedure of keeping score? (They can capture the user's attention and alleviate the drudgery or boredom that sometimes accompanies drill routines.)

- So far the characters that we have drawn on the text screen have appeared to move by floating on the screen. This is due primarily to the fact the we are drawing, erasing, and redrawing the same figure. Is there a way to simulate movement in a more realistic manner? (Redraw the same figure with specific body parts in different positions.)

Aim: What are some of the ways that we can program our computers to achieve realism when simulating moving bodies?

Development:

1. An advertisement for snack food boasts, "Bet you can't eat just one." We can reword this advertisement and make it apply to a video game. "Bet you can't play just once." If we could capture some of the enthusiasm that students feel when playing a video game and incorporate it into the programs that we write, we would provide our classmates with an entertaining way of developing basic skills. What are some of the factors that appeal to the user of a video game?

- Direct student discussion to establish the following:

  - The user is challenged to beat his previous score.

  - The graphic display adds to the excitement.

  - A maze may have been involved and the user wants to attempt a new course.

  - Some reward may have been offered and the user is sure of obtaining the reward at the next try.

  - The user finds her skill improving as she goes along and is convinced she will be successful at the next try. The user is aware of the mistakes that she made the first time and wants an opportunity to rectify her errors.

- It should now be established that any challenge that makes a user repeat a drill can only end in improving that person's skills.

- Keeping score makes the user aware of his performance and challenges him to improve. Develop a list of methods for keeping score.

  - Start with zero and add a given number of points for the correct answer. These points can vary from question to question, or be of equal value.

  - Start with any given number, add points for a correct answer, deduct for an incorrect one. Use the same number in both instances or weight the correct answers more heavily.

  - Award points on the basis of the number of tries needed to achieve the correct answer. Five points for success at the first try, four points for the second, etc.

  - Develop a list for ways of displaying this score.

    .A score box in the corner of the screen.
    .A figure climbing up stairs with a reward at the top.
    .A basketball that passes through a hoop every time a question is answered correctly.
    .A baseball player that completes a home run.
    .A moving stream of ducks, as in a shooting gallery. One disappears with every correct answer.
    .A moving figure that walks to a scoreboard and then changes the total.

2. Write a segment of a program that will keep score by drawing an alien on the bottom step and a flying saucer at the top step. Challenge the user to help the alien reach his saucer.

Draw figures:

```
                                                  ___
                                                 ( )
                                                /UFO|
                                               /ooooo|
                                              ________
                                               _______
                                                !
                                                !
                                          _______
                                          !
                                          !
                                    _______
                                    !
                                    !
                              _______
                              !
                              !
                        _______
                        !
                        !
                  _______
                  !
                  !
            _______
            !
            !
      _______
      !
      !
  ^
 / |
/o o|
/ o |
/ - |
 XXX
 XXX
 XXX
 L L
_______
i
!
```

HELP THE ALIEN REACH HIS FLYING SAUCER.
FOR EACH CORRECT ANSWER ADVANCE TWO STEPS.

. Plan the program:

- Determine the number of questions, relate to the number of steps.

- Investigate defining cartoon parts as strings.

- Strings work very well when erasing the alien. A simple FOR/NEXT loop with a string of blanks is all that is needed.

- Care must be taken when calculating screen position so as not to erase any part of the steps.

- Determine if you are going to permit a negative score. This can be accomplished by erasing the alien completely and not showing him again until there is a positive score.

- Plan on informing the user how many questions she answered correctly and how many she missed. Provide for two counters. Determine how many opportunities the user will have to arrive at the correct answer before you either tell her the answer or advance to the next question.

- Establish a variable to be used as an accumulator. Use simple commands such as:

```
IF A$ = "FRANCE" THEN A = A + 2:C = C + 1
IF A$ <> "FRANCE" THEN A = A - 1:K = K + 1
```

- Relate A to screen column for drawing alien. This is easily accomplished with two different subroutines. First, erase him where he is and then branch to a subroutine that draws him one step backward, or a subroutine that draws him two steps forward.

. Write a program and save it to be used as part of a remedial practice program.

3. Earlier in the lesson we discussed a way to simulate movement with more realism than we had in the past. I wish to write a program that will show a man walking across the screen. What would be an effective way of simulating this movement?

. Establish the need to draw more than one picture.

. Discuss the parts of the body that move when a person walks.

. Arms and legs spread out and then appear to return to the body.

. Agree to draw three different positions.

. Select keys and draw figures.

```
 ^^^
 A  o)
 ?   >
 !   <
  | /
   X
   !
  /!|
 / ! |
c  !  c   Position 1
   X
  / |
 /   |
/     |
L      L
```

```
 ^^^
 A  o)
 ?   >
 !   <
  | /
   X
   !
  /!|
  /!|
 c ! c    Position 2
   X
  / |
  / |
  / |
  L  L
```

```
^^^
^  o)
?   >
!   <
 | /
  X
  !
  !
  !
  !      Position 3
  X
  !
  !
  !
  L
```

. Plan program:

- Draw each position as a subroutine.

- Begin and end with position 2.

- Position 3 must be flanked on either side by position 2.

- Position 1 must be flanked on either side by position 2. Agree on the following sequence: 2, 1, 2, 3, 2, 1, 2.

- Since there is no other material on the screen, we can erase each figure by clearing the screen.

- A pause must be used to keep the figure on the screen before it is cleared. The pause does not have to be the same for each figure. Agree to make the initial pause the longest and all other pauses of equal length.

- Calculate screen columns and determine cursor positions for drawing each figure.

- Suggested program for computers that support HTAB and VTAB.

```
5   REM WALKING MAN
6   HOME
10  X = 1:Y = 1: GOSUB 310
15  FOR T = 1 TO 300: NEXT T: HOME
20  Y = 6: GOSUB 100
25  FOR T = 1 TO 200: NEXT T: HOME
30  Y = 10: GOSUB 310
35  FOR T = 1 TO 200: NEXT T: HOME
40  Y = 15: GOSUB 510
45  FOR T = 1 TO 200: NEXT T: HOME
50  Y = 20: GOSUB 310
55  FOR T = 1 TO 200: NEXT T: HOME
60  Y = 25: GOSUB 110
65  FOR T = 1 TO 200: NEXT T: HOME
70  Y = 30: GOSUB 310
75  FOR T = 1 TO 200: NEXT T
```

```
 80  END
100  VTAB X
110  HTAB Y: PRINT " ^^^^"
120  HTAB Y: PRINT " ^  O"
130  HTAB Y: PRINT " ?   >"
140  HTAB Y: PRINT " !  -<"
150  HTAB Y: PRINT "  | /"
160  HTAB Y: PRINT "   X "
170  HTAB Y: PRINT "   ! "
180  HTAB Y: PRINT "  /!|"
190  HTAB Y: PRINT " / ! |"
200  HTAB Y: PRINT "C  !  C"
210  HTAB Y: PRINT "   X"
220  HTAB Y: PRINT "  / |"
230  HTAB Y: PRINT " /   |"
240  HTAB Y: PRINT "/     |"
250  HTAB Y: PRINT "L      L"
280  RETURN
310  HTAB Y: PRINT " ^^^^"
320  HTAB Y: PRINT " ^  O"
330  HTAB Y: PRINT " ?   >"
340  HTAB Y: PRINT " !  -<"
350  HTAB Y: PRINT "  | /"
360  HTAB Y: PRINT "   X "
370  HTAB Y: PRINT "   ! "
380  HTAB Y: PRINT "  /!|"
400  HTAB Y: PRINT "  /!|"
410  HTAB Y: PRINT " C ! C"
420  HTAB Y: PRINT "   X"
430  HTAB Y: PRINT "  / |"
440  HTAB Y: PRINT "  / |"
450  HTAB Y: PRINT "  / |"
460  HTAB Y: PRINT "  L L"
490  RETURN
510  HTAB Y: PRINT " ^^^^"
520  HTAB Y: PRINT " ^  O "
530  HTAB Y: PRINT " ?   >"
540  HTAB Y: PRINT " !  -<"
550  HTAB Y: PRINT "  | /"
560  HTAB Y: PRINT "   X "
570  HTAB Y: PRINT "   ! "
580  HTAB Y: PRINT "   ! "
590  HTAB Y: PRINT "   !  "
600  HTAB Y: PRINT "   !
610  HTAB Y: PRINT "   X "
620  HTAB Y: PRINT "   ! "
630  HTAB Y: PRINT "   ! "
635  HTAB Y: PRINT "   ! "
640  HTAB Y: PRINT "   L "
670  RETURN
```

4. How can a walking man be used in a drill program?

   . The rightmost columns of the screen can be used to keep score. Each time our figure reaches the end of the screen the score can be changed.

- Since the screen is cleared every time the figure appears, this would be a good way of clearing the screen between questions.

- The user can be given the opportunity of trying different responses until the correct answer is achieved. The screen can then be cleared for a new question and the score advanced.

- If the question area is kept to the right of the screen, the final position of the man and the score can remain on the screen while the question is being answered.

- When the final question is answered correctly, a flashing "success" can be printed.

- The walking man not only provides a way of clearing the screen, but it acts as a diversion from the routine of the drill.

Applications:

1. Write a program that will cause ducks to float across the screen. Provide for seven ducks on the screen. Assume every correct answer will flatten a duck. Stop the movement of the ducks at regular intervals to await user input. Write your program so that when the letter "Y" is pressed a duck is flattened. Pressing any other key will have no effect on the program. End program when there are seven flattened ducks. Use strings to define the figures and take care to position the cursor in the correct screen column.

   - Suggested figures

```
 ^
>o>
 $
>( )
 V-V
```

```
 *
^^^
```

2. Use the star key to spell out the words TRY AGAIN. Make each letter occupy five columns and five screen lines. Use the flying saucer we developed earlier in the lesson to slowly trail this phrase across the screen. Simulate realism by not keeping the saucer in the same position relative to the letters. The letters should move at a steady rate but the saucer should hover on the screen for different time intervals and at random screen positions.

Summary:

1. How does keeping and displaying a score challenge the user?

2. What is the difference between a counter and an accumulator?

3. When does the random function aid in providing an effective graphics display?

4. What is the effect of introducing moving graphics in a program that attempts to improve a student's basic skills in a particular subject?

5. Are there specific areas of a drill program where moving graphics would detract from our purpose?

6. What are some of the tricks we use to simulate realistic movement when we write programs?

7. What are some of the rewards we can offer a student for the successful completion of a program?

Assignmments:

1. Modify the "Stairs" program to simulate a dog trying to reach a bone. Use the suggested dog below or create one of your own. Be even more creative and use two other figures of your choice. You may redesign the point allocation and the control keys, or use the same ones we developed in class. Suggested dog:

```
     "
|    o>
 OOOOO
  M M
```

2. Simulate the movement of a snake crawling across the screen. Draw the snake in two or three different positions as we did with the moving man program. Use the suggested snake or create a snake of your own. Suggested snake:

```
            ^
O          o*o     For another position draw body flat on ground,
 O   OOO   Y      for another draw body like an inverted "V".
  OOOO OOOO
```

3. Design a scoreboard similar to those used at ball games. Provide areas for NUMBER RIGHT, NUMBER WRONG, TOTAL SCORE. Change the displays by using specific keys. For example, if the "R" key is pressed the number right advances one digit. Allocate five points for each correct answer and deduct two for each error. Outline the board with a moving stream of lights. This can be accomplished using the same procedure that we used to imitate an electrical current.

Assignment for More Advanced Students:

1. Write a program that imitates a baseball diamond and a player advancing along the bases. Provide for keyboard input. Use four different keys to position the player at the various bases. In all cases show his path from home plate to the allocated base.

Topic 12
APPLE MEMORY LOCATIONS

(For Apple users only)

Performance Objectives: Students will be able to:

- understand the Apple II memory map
- look at any memory location in the Apple II computer via the PEEK command
- change any memory location in the Apple II computer via the POKE command

Vocabulary: New: PEEK, POKE
Review: RAM, ROM, ASC, CHR$

Motivation:

- Type in the following program. Before running it, however, can you predict what it will do?

```
10  HOME
20  POKE  - 16298,0
30  FOR I = 3 TO 72
40  POKE  - 16304,0
50  FOR J = 1 TO 10: NEXT
60  POKE  - 16303,0
70  NEXT
```

Aim: How are PEEK and POKE statements useful to the computer programmer?

Development:

1. How many types of memory does a computer have?

   Answer:

   There are two types of computer memory, RAM and ROM. RAM stands for random access memory. It is the space where the computer stores programs and other data. Material in RAM can be erased or changed. ROM stands for read-only memory. This information is stored in a computer and in general cannot be changed.

2. How can we look directly at any part of the computer's memory?

   Answer:

   In BASIC, the PEEK command returns a decimal value of the memory location specified:

   PRINT PEEK(800)

   This will print the decimal value found in location 800. (Either a RAM or ROM location can be read.)

3. How can we change a memory location in the computer?

Answer:

In BASIC, the POKE command will allow a change in a RAM location:

POKE 800, 45

This will write the value 45 to location 800. (Only a RAM location can be written to.)

4. Where is the computer's memory?

Answer:

The RAM and ROM of the computer is located in the I.C. chips inside the computer. Everything the computer can store, process or do is located in these memory locations.

5. What does each memory location do?

Answer:

Essentially memory locations store values. Some of these locations store the program, others store the instructions to convert the program into machine code (BASIC), other locations control what is shown on the screen, while others store switch values such as whether to show text or graphics. Different computers store information in different areas of their memory, but essentially the process is the same.

6. What's where in the Apple II? (There are many books on the market which include memory maps and go into computer memory in greater depth. Such knowledge is very important to the programmer's understanding of what is happening inside the computer.)

(Each memory location is called a BYTE, and can store a value of 0-255. A 64K machine, like the Apple IIe, consists of 65,536 BYTES of memory.)

7. Each location can have a positive or negative value by adding or subtracting the number 65,536. For example, the BYTE 49232 is the same as -16304.

8. Why are PEEK and POKE commands important? (They provide greater programming power and control over the computer.)

9. Where can we POKE some values without causing problems? (Programmers like to use the area 768-1023 to POKE values or short machine language routines -- see lessons on generating sound.)

Applications:

1. One advantage of POKEing information into memory is that it will not be changed by loading a new program. Programmers use this to pass information from one program to another.

2. Write a program that will ask for the day of the month and POKE it into location 768. Then write another program that will read this memory

location and print out the day of the month.

3. How can characters be stored in memory? (By using the ASC(A$) of that character.)

   Example:

```
10 PRINT "TYPE A CHARACTER ";:INPUT A$
20 X= ASC(A$): REM ASC OF A TYPED CHARACTER
30 POKE 768,X
```

   To print this memory value:

```
10 X= PEEK(768)
20 PRINT CHR$(X)
```

Summary:

1. How can we look at any memory location in our computer?
2. How can we change any memory location in our computer?
3. Which type of memory can we change?
4. What are the memory locations called?
5. How many memory locations can a 64K computer have?
6. Why are POKEing and PEEKing important?

Assignments:

1. Write a program that will POKE values 1 to 100 directly into location 768 to 868.
2. Write a program that will READ and PRINT the values found in memory location 768 to 868.
3. Write a program that asks the user to supply three numbers. Find the sum of these numbers and poke the answer into location 1000. Have the program do this four more times, storing each answer in locations 1001 to 1005 respectively.
4. Write a program that will read the values of locations 1000 to 1005 and find the average of these.

Assignment for More Advanced Students:

1. Write a program that will ask for today's date. Then poke that response into the computer's memory. Then write another program that can read that information from the computer's memory.

Topic 13
USING PEEK AND POKE ON AN APPLE COMPUTER

(For Apple users only)

Performance Objectives: Students will be able to:

- use PEEK and POKE statements in order to format the text screen
- use PEEK and POKE statements to read the keyboard
- learn some other useful PEEK and POKE locations

Vocabulary: Review: PEEK, POKE, VTAB, HTAB, FOR/NEXT

Motivation:

- Type in the following program. Before running it, however, can you predict what it will do? Write your prediction before running this program. Then, describe what the program did.

```
10  TEXT : HOME
20  VTAB 3: HTAB 5
30  PRINT "SEE THE NUMBERS:"
40  PRINT "*********************"
50  POKE 34,4
60  POKE 35,20
70  POKE 33,20
80  FOR I = 1 TO 1000
90  PRINT I;
100  NEXT
```

Can you explain why things happened the way they did?

Aim: How are PEEK and POKE statements able to control the text screen, keyboard and other functions?

Development:

1. How can we control text output with POKE statements? (As we saw in the motivation, we were able to control the text output by POKEing various numbers into the proper locations.)

   The table below lists the memory locations and functions they control:

Memory Location	Function	Normal Value
32	left margin of text window	0
33	sets width of text window	40
34	top of text window	0
35	bottom of text window	24

2. Can you give an example of controlling text output to enhance programming?

Acceptable answer:

We can print our program title on the first text line, then lower the top of our text window to protect it from scrolling off the screen, or being erased by the HOME command.

```
10 TEXT:HOME
20 VTAB 1
30 PRINT "THIS IS MY PROGRAM:"
40 POKE34,2
50 FOR I=1 TO 1000
60    PRINT I, I*2, I*4
70 NEXT I
```

3. The VTAB command is not bounded by the text window and therefore allows us to jump around our text window. As an example, add the following line to the above program:

```
65 IF I=900 THEN HOME: VTAB1: PRINT "ALMOST DONE": VTAB 2: REM IN AND OUT
   OF WINDOW
```

4. PEEK and POKE statements can be used to read the keyboard and produce a value 128 larger than the ASCII value of the last key pressed before the peek was made. This keypress can be cancelled with POKE 49168,0. Note the character can be converted into a screen character by then using the CHR$ command.

```
5 REM KEYBOARD STROBE
10 PRINT "PRESS ANY KEY"
20 X = PEEK (49152) - 128: REM READ KEYBOARD
30 IF X <32 THEN GOTO 20
40 POKE 49168,0 : REM CLEAR KEYBOARD
50 PRINT "YOU PRESSED THE :"; CHR$(X)
60 GOTO10
```

5. How can we restore the text window to the normal positions? (Type TEXT in immediate or deferred modes, or perform a RESET.)

Applications:

1. How can we use the Keyboard Strobe routine to create a time limit to answer a question?

```
10 PRINT "WHAT IS THE SUM OF 3 + 2?"
15 FOR I = 1 TO 3000
20    X = PEEK (49152) - 128: REM READ KEYBOARD
30    IF X = 5 THEN PRINT "RIGHT!":END
35 NEXT I
40 POKE 49168,0 : REM CLEAR KEYBOARD
50 PRINT "SORRY! TOO LATE!"
60 END
```

2. How can we modify the Keyboard Strobe program so that only a letter answer will be permitted?

Acceptable Answer:

Look at an ASCII chart to see that the ASCII values for the alphabet are greater than 64.

Therefore replace line 30 with:

```
30 IF X <64 THEN GOTO 20
```

3. Other useful locations:

Memory Location	Function
22	Error Code
36	Horizontal Cursor Position
37	Vertical Cursor Position
43624	Drive Number
43626	Slot Number
46064	Number of sectors per track

Summary:

1. How are POKE and PEEK statements important to the programmer?

2. How can we control the text screen of the Apple II with POKE statements?

3. How can we read the keyboard directly for input?

4. How can we eliminate inappropriate responses with our Keyboard Strobe routine?

Assignments:

1. Write a program that will strobe the keyboard and will only print numbers.

2. Use the Keyboard Strobe routine as a subroutine in a program which asks three questions.

3. Adjust the text windows so that the numbers from 1 to 1000 are printed in a long narrow column in the center of the text screen.

4. Write a program that will PEEK and PRINT the values 0 to 255 on the computer.

Assignments for More Advanced Students:

1. Write a game program that will pick a character at random and allow you only 10 seconds to match that character by pressing keys on the keyboard.

2. Using the Keyboard Strobe routine, make a character move right or left or up or down depending on which arrow keys are pressed. Keys other than the arrow keys may be used.

Topic 14
USING SOUND AS A VEHICLE FOR INSTRUCTION

Performance Objectives: Students will be able to:

- create a list of criteria for the use of sound in a program
- understand how sound enhances a teaching program
- learn what types of sound they can produce, using their computers
- develop an appreciation for the use of music as a motivational device
- examine the component parts of a song and determine what must be considered before sound can be turned into music
- examine the effectiveness of different types of sounds on different types of programs
- suit a particular song to a specific program, or a specific portion of a program

Vocabulary: New: pitch, meter, duration, scale, rest
Review: teaching program, motivational device, pause

Motivation:

- Create a very loud sound that will startle the class and then react by saying: Is anything wrong? Why are you all looking at me like that?

- Develop the following statement:

  - The introduction of sound is a useful tool in capturing the attention of an audience.

- Ask the following question: If I began every lesson with the same sound, what would some of your reactions be?

  Attempt to elicit the following responses:

  - Some students would cover their ears while others would hide under their desks in an effort to avoid an unpleasurable experience.

  - After a day or two it would have little or no effect on the majority of the students.

  - It would encourage lateness.

- How can we create sound that is pleasurable?

  - Imitate the sound of an instrument.

  - Simulate music.

  - Sing a song. Offer to sing the class a song and ask them what they would like to hear. (Consider yourself lucky if you recognize the names of any of their requests.)

  - Inform them, in a quiet voice, that you think you will sing them Brahms' Lullaby. (This should enliven a discussion of the suitability of the selection.)

Aim: When is sound an effective tool for enhancing a teaching program, and what considerations are necessary when selecting that sound?

Development:

1. What kinds of sounds can be created on a computer?

   . A bell. (Usually accomplished by pressing a given key or a series of keystrokes.)

   . An audio click. (Usually the result of tweaking the speaker within the computer and accomplished by some machine language routine.)

   . If either of the above are programmed into a FOR/NEXT loop, a series of these sounds results. Varying the pause between the clicks or the bells results in other types of sounds.

   . Most computers employ machine language routines to create music. Composers are writing computer symphonies and an abundance of software is available for writing music on your computers. Some children's games are available that turn the computer keyboard into a piano. Many video games are accompanied by very realistic sounds and some are introduced with a song.

   . If it is possible, this would be a good time to demonstrate the effect of an auxiliary speaker. If a tape recorder with a good speaker could be hooked up to your computer, you could demonstrate the change in volume. Most computers do not come with volume controls, and in a large classroom students may have to strain to hear the computer's speaker. Amplifying the sound would therefore serve a double purpose.

2. Before continuing with the lesson, it is wise to define any vocabulary that may be new to some of the students. There is a large segment of the population that consider themselves tone deaf. It is surprising how many students feel uncomfortable when faced with the possibility of having to create music.

   . Pitch refers to the note. A low pitch is found at the bottom of the scale, high pitch at the top.

   . Scale usually refers to the Bach scale, (8 tones and 12 semitones). We attempt to match the pitch as closely as possible. Most computers do a fine job (that is if we consider the relation of the sounds to each other). If you are striving for perfection or trying to match the computer with your favorite vocalist, you might be in for a disappointment.

   . Meter refers to the timing. Music is created not only by varying the pitches, but by holding the notes for different lengths of time.

   . Duration refers to the length of time that a particular note is held.

   . Rests (as the word implies) are spaces within a song where no music is heard.

3. Develop a list of situations where sound can enhance a program. In this lesson we are mainly concerned with teaching programs. Begin the discussion with the use of sound as a motivational tool. A peppy routine such as a bugler announcing roll call, suggests a method of perking up a student's attention for an additional drill practice. This topic can be divided into three sections. How can we motivate students to begin a program, continue with a program, and return to a program for additional practice?

4. What kind of sound would you create to accompany a program's introduction?

   - A popular song that the student is familiar with will relax him and make him feel at ease.

   - A sound that begins with a low pitch and seems to increase not only in pitch but in intensity as well.

   - The sound of a siren which is created by moving up the scale very rapidly, while at the same time simulating an increase in volume, is an effective way of capturing the attention of the user.

   - A simulated, mild explosion accompanied by a graphic display is a better way to begin than to end. It wakes the user up without frightening her (we hope) and keeps her attention on the screen. These explosions should be of the 4th of July fireworks type rather than bombing and devastation.

5. What type of sounds would you use during a program?

   - A simple bell or razz to announce an error.

   - Imitate the sound of an automobile horn to signal, "Watch out, think about what you are doing."

   - A collection of bells ringing to spell success.

   - A catchy tune such as "We're in the Money" is enough to make a user smile at her own competence. These tunes do not have to be played with every correct answer but can be programmed to be played whenever the score advances a given number of points.

   - A tune such as "Merrily We Roll Along" could accompany the turning of a page.

   - Bugle calls are great waker-uppers.

6. What kinds of sound would you use to terminate a program?

   - Descending sounds that vary from high to low pitch, whose volume seems to fade.

   - Goodbye music such as "The Party's Over" or "Olde Lang Syne." Programs can be written to cause the computer to sign off with different songs, depending on the range of the user's score.

. "High Hopes" is a good song to end with. It almost promises the user that he will do better the next time.

7. Can we write a BASIC program that will cause the computer to play a tune? (Yes, but we do not have instructions in BASIC to play music. We must rely on machine language routines that are poked into the computer's memory and then called up by our BASIC program.)

   . What are some of the factors that must be taken into consideration when planning this BASIC program?

     - The area of the computer's memory where the machine language program that creates the music will be stored.

     - The pitch of each note.

     - The duration of each note.

     - Where rests occur.

     - How the BASIC program will store this information.

   . What are some of the BASIC commands to handle these procedures?

     - After the machine language routine is written it can be stored in an array. FOR/NEXT loops and READ/DATA statements can be used to POKE the routine into the computer's memory at the designated location.

     - After the number of notes in the tune is determined, arrays can be dimensioned and filled. Use two separate arrays, one for note and the other for the duration. Again FOR/NEXT loops and READ/DATA statements are used to POKE these values into the machine's memory.

     - The procedure that we use is as follows: POKE a note and a duration into the machine's memory, CALL on the machine language routine and the note will be played at the given duration. Repeat this procedure for the next note and continue in this manner until all notes are played.

     - When a rest is required branch to a subroutine that creates a pause.

8. Develop a list of tunes that suit particular programs. When selecting a tune it is generally a good idea to avoid love songs, hymns, and inspirational music. Some suggestions follow:

   . Programs that begin with graphics displays lend themselves to particular tunes.

     - Begin a program with a large music box playing "Pop Goes the Weasel." Imitate the action and sound of an organ grinder. Have the box open and the name of the program emerge from the box as a Jack in the Box would. By using a very short duration for each note and repeating the note several times, the sound of an organ grinder can be simulated.

     - Programs that begin with graphic displays of rockets, stars and

planets can be accompanied by the "Star Wars' Theme" or the theme from "Star Trek."

- "Good Morning" and "Oh What a Beautiful Morning" are good tunes to accompany an initial floral display, or a landscape.

- A program on some aspect of the weather can be accompanied by "Singing in the Rain."

- A spelling practice can begin with the alphabet song.

- A program that orients freshmen to your school might play a school song.

- One that provides SAT review might play a popular college song. (Cornell, Yale, Columbia)

- Programs that are written like games, where scores are displayed as home runs, can be accompanied by "Take me Out to the Ball Game."

Applications:

1. Tap out the meter for "Pop Goes the Weasel." Allocate a duration of 25 for the shortest note and 100 for the longest note. Write a segment of a song program that dimensions, saves, and reads an array for the duration of each note in one stanza of this song.

   Acceptable answer:

```
10 REM SET NUMBER OF NOTES
20 N = 29
30 REM DIMENSION ARRAY
40 DIM M(N)
50 REM SAVE DURATION IN ARRAY
70 READ M(D)
80 NEXT D
90 DATA 50,50,50,50,25,25,25,50 50,50,50,50, 50,100,100,50,50,50,
  50,50,25,25,25,100,100, 50,50, 100,100
```

2. Clicking sounds, ones that imitate Geiger counters, are easy to create on computers. Name five situations that occur in teaching programs that would benefit from the introduction of this type of sound.

3. Name a song that could be used in an introductory module to the following programs.

   . A social studies lesson that asks the user to locate cities on a state map.

   . A program that provides remedial practice in basic mathematics skills.

   . A French vocabulary lesson.

   . An examination of several aspects of constitutional law.

   . A biology lesson that examines the different parts of a flower.

4. What type of a program can be merged with the following songs?

   "Hail to the Chief"
   "My Merry Oldsmobile"
   "Old Man River"
   "Ding Dong the Witch is Dead"
   "God Bless America"
   "Theme from 2001"
   "Row, Row, Row, your Boat"
   "The Yellow Rose of Texas"
   "There's No Business Like Show Business"

Summary:

1. What types of sounds is the computer capable of making?

2. Where in a program would music detract from the program's purpose?

3. How do computers create music?

4. Would a shrill note have a high or a low pitch?

5. What is the purpose of setting a duration for each note?

6. If we POKE the duration of a note into the computer's memory and our computer can only receive a number named by two hexadecimal digits, then what is the longest duration that we can use in a BASIC program?

7. What kind of sound can we use in a program to inform the user that he has just made a mistake?

8. If you heard a computer playing "O Solo Mio" what kind of a program would you guess the student was using?

9. What would be a good song to merge with an automobile simulator program? A flight simulator?

10. Why is music such a great motivational device?

Assignments:

1. Modify the program we wrote in class that saved in an array the meter for "Pop Goes the Weasel," to save the meter for "Happy Birthday." Use 50 as your shortest duration and 200 as the longest duration.

2. Many songs have been written that we associate with great leaders of our country. Who do you think of when you hear:

   "The Battle Hymn of the Republic"
   "The Fife and Drum Song"
   "Happy Days are Here Again"
   "The Missouri Waltz"

3. Other songs are associated with periods in history. What eras do the following songs bring to mind?

"Brother Can you Spare a Dime?"
"Over There"
"We Shall Overcome"
"Waltzing Matilda"

4. Most countries have national anthems. These songs are handy for history and language lessons. Make a list of all those that you know.

5. A program tests the speed at which a student reads a paragraph and then answers three questions. When the last response is recorded the student is shown the elapsed time.

   . Where in the above program would you make use of sound?

   . When would it be unwise to use sound?

   . Assume that the program refused to accept wrong answers and the student was forced to make second, third and fourth choices. Would this be an opportunity for introducing sound?

   . Assume that the student selected all three correct answers. Would this affect your choice of sound?

6. You are writing a program for your English teacher. The aim of the lesson is to find as many grammatical and spelling errors as exist in a planned paragraph. You program a paragraph with 10 errors. Is there any place in this program for sound?

Topic 15
SOUND GENERATION—IBM

Performance Objectives: Students will be able to:

. understand how the BEEP, SOUND and PLAY statements work

Vocabulary: hertz, BEEP, SOUND, PLAY

Motivation:

. How can the computer be programmed to play, "Twinkle, Twinkle Little Star?"

Aim: What do the BEEP, SOUND and PLAY statements do?

Development:

1. There are three BASIC statements which generate sound on the IBM PC. They are: BEEP, SOUND, and PLAY.

2. The BEEP statement produces a sound at only one frequency (800Hz) and lasts 1/3 second. Examine the following:

```
10 PRINT "BEEP"
20 BEEP
30 END
```

   What is the result of running this program?

3. To generate sound using the SOUND statement, the frequency and duration must be specified. The range of values for the frequency is 37Hz to 32,767Hz. A person can normally hear frequencies ranging from approximately 37Hz to as high as 15,000Hz. The range of values for the duration is 0 to 65,535 clock ticks. Eighteen clock ticks is about one second. The general form of the SOUND statement is:

```
120 SOUND frequency, duration
```

   For example:

```
250 SOUND 440,36
```

   will produce the A note for approximately two seconds.

   A SOUND statement with a duration of 0 cannot be heard.

4. The SOUND statement can be used in two modes: music foreground mode and music background mode. In the music foreground mode, a sound generated by a SOUND statement must finish before another sound can be produced by the next SOUND statement. In the music background mode, the computer will store subsequent sounds to be generated in a buffer. After the current sound has ended, the computer will retrieve the sound stored in the buffer and generate it. This allows the computer to continue executing the program. The maximum number of sounds that can be stored is 32.

5. The PLAY statement can be used to produce tunes on the IBM PC in <u>Advanced BASIC only</u>. The general form of the PLAY statement is:

PLAY string

where the string consists of characters, defined by IBM's "tune definition language." These commands are:

A to G	notes in the current octave.
#, + or -	# or + represent a sharp. - represents a flat. These symbols can be used only for notes that correspond to a black key on the piano. For example, C flat is not allowed, since it corresponds to B, a white key on the piano.
O x	Octave. x can be a value from 0 to 6 corresponding to seven octaves. Each octave starts with C and ends with B. Octave 3 is the octave beginning with middle C. The default octave is octave 4.
L x	Length of notes following it. x ranges from 1 to 64. For example:

Length	Equivalent
L1	whole note
L2	half note
L4	quarter note
L8	eighth note
L16	sixteenth note

P x	Pause (rest). x ranges from 1 to 64 and is the same as length in terms of duration.
.	dot or period. If it follows a note or a pause, it causes the note to be played as a dotted note.
T x	Sets the number of quarter notes per minute. x ranges from 32 to 255. The default value is 120.
MF	Music foreground.
MB	Music background.
MN	Music normal. Causes each note to be played 7/8 of the time specified by the length (L).
ML	Music legato. Causes each notes to be played the full amount of time specified by the length.
MS	Music staccato. Causes each note to be played 3/4 of the time specified by the length.

6. For example, examine the following program:

```
10  PLAY "O3 G8 A8 B O4 D D. E8 D O3 B G A B B A G"
20  PLAY "O3 A2. G8 A8 B O4 D D. E8 D 03 B G A B B A A G2."
30  PLAY "O3 C2 C2 E E2 E D D O2 B G A2. P4 O3 G8 A8"
40  PLAY "O3 B 04 D D. E8 D O3 B G A B B A A G2."
50  END
```

What is the result of running this program?

Applications:

1. Write a program which will play, "Merrily We Roll Along" or "Mary Had a Little Lamb." (see diagram 1)

Merrily We Roll Along

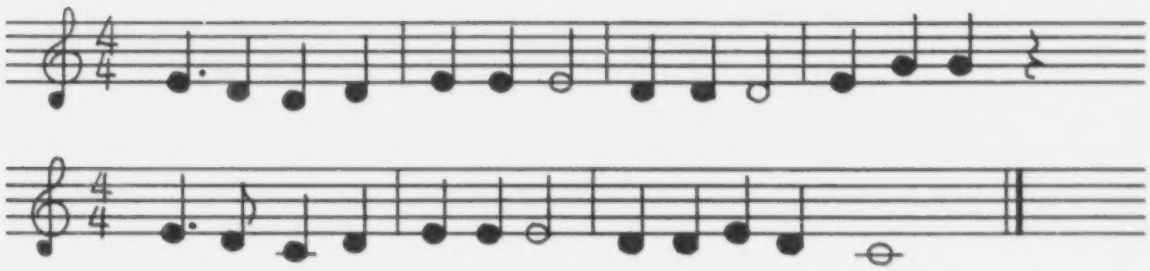


Diagram 1

2. Write a program that will generate simple two-digit addition problems and beeps when the user types in the wrong answer.

Summary:

1. What can the BEEP statement do?

2. What are the differences between the SOUND and PLAY statements?

Assignments:

1. Write a program which will generate multiplication problems for the user to answer and cause the computer to play a little tune when the answer is right and to beep when the answer is wrong.

2. Write a program which will cause the computer to play, "Twinkle Twinkle Little Star." (See diagram 2)

Twinkle, Twinkle Little Star

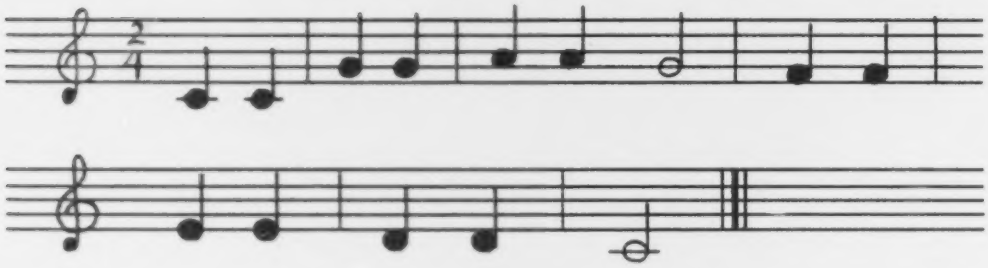
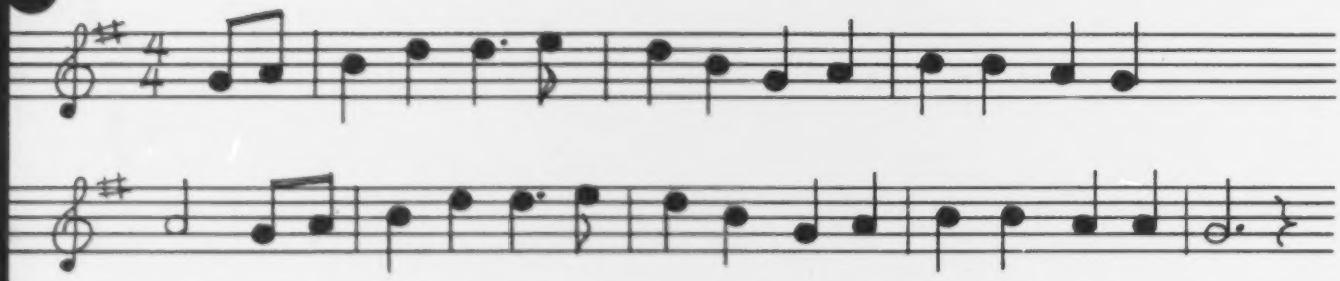


Diagram 2

3. Choose your own piece of music and write a program that will cause the computer to play it.

Oh! Susanna



(Music for Ex.)
Play Statement

Topic 16
SOUND GENERATION—APPLE—PART I

Performance Objectives: Students will be able to:

- program the computer's speaker to produce musical tones
- learn how to install and use a machine language sound routine into computer memory from a BASIC program
- understand the effects of PITCH and DURATION on musical tones

Vocabulary: New: pitch, duration, CALL, installed
Review: POKE, PEEK, GOSUB, FOR/NEXT

Motivation:

- The Apple II uses memory location -16336 as a sound switch. Each time you look at this location in BASIC with a statement such as X=PEEK (-16336) a click is produced from the speaker.

- Write a short BASIC program to click the speaker 10 times with a GO TO loop with an incremented variable to end the program on the 100th loop.

- Write a short BASIC program to click the speaker 100 times using a FOR/NEXT loop.

- Change the FOR/NEXT loop to click the speaker 100 times and note how this affects the quality of the sound.

Aim: How can we program the computer to generate musical tones?

Development:

1. Using a BASIC program to drive the speaker (as we did in the motivation) is easy to do but the type of sound generated is very limited.

   - The manner in which the computer processes the BASIC programming instructions is relatively slow, producing low frequency tones.

   - The faster we can pulse or cycle the speaker the higher the pitch will be.

   - The pitch of the tone produced will depend on the relative number of times (frequency) we pulse the speaker.

   - The length of time a particular tone is played also affects the quality of that note. This is called the duration. The greater the duration the more intense that note will be.

2. A machine language routine can be used to produce tones with greater ranges.

   - Why does a machine language routine run faster than a BASIC routine? (Instructions do not have to go through the BASIC interpreter)

   - One way to place machine language instructions into the memory of the

computer is to POKE the decimal values of the machine codes into an appropriate area of memory.

- Many programmers prefer to use the memory area located between 768 and 1023 ($300 - $3FF in hexidecimal) for short machine language routines.

3. Type the following into a BASIC programming statement:

```
100 POKE 770,173: POKE 771,48:  POKE772,192:  POKE 773,136
110 POKE 774,208: POKE 775,5:   POKE 776,206: POKE 777,1
120 POKE 778,3:   POKE 779,240: POKE 780,9:   POKE 781,202
130 POKE 782,208: POKE 783,245: POKE 784,174: POKE 785,0
140 POKE 786,3:   POKE 787,76:  POKE 788,2:   POKE 789,3
150 POKE 790,96
```

4. Once the above program has been run, we may then consider the sound routine as being "installed" into the computer's memory. As long as the computer is not turned off, this routine can be accessed by any program which correctly uses the "installed" routine.

5. How can we produce tones? The following is a program that will allow us to produce tones:

```
10 P=100: REM P IS THE VALUE OF PITCH 1-255
20 D=125: REM D IS THE DURATION 1-255
30 POKE 768,P
40 POKE 769,D
50 CALL 770
60 END
```

6. Any integer value from 1 to 255 may be used for the P and the D variables which are the pitch and duration values respectively.

Applications:

1. In order to observe the effect of pitch upon the tone, modify the program above so that the duration remains 100 (D=100) but increment the value of P by fives from 5 to 255. What is the effect upon the tone by increasing the value of P?

2. In order to observe the effect of duration upon the tone, modify the program above so that the pitch remains 100 (P=100) but increment the value of D by fives from 5 to 255. What is the effect upon the tone by increasing the the value of D?

3. How can we change our program into a subroutine? (Change line 60 RETURN)

4. Using a FOR/NEXT loop, create a subroutine which will play tones from 1 to 255 with a constant duration of 10.

Summary:

1. How can we click the speaker from BASIC?

2. How can we drive the speaker at faster speeds?

3. Why do we need a machine language routine for sound?

4. How can we enter a machine language program from BASIC?

5. How can we produce desired tones once a driver routine is installed?

6. What effect does pitch and duration have upon the tone?

Assignments:

1. Write a subroutine that will poke the machine language subroutine into memory using a FOR/NEXT loop with READ/DATA statements.

2. Write a program that will produce ten tones increasing in pitch.

3. Write a program that will produce the same note ten times, each time increasing in duration.

4. Write a program that will produce three notes increasing in pitch followed by three notes decreasing in pitch all with different durations.

Assignments for More Advanced Students:

1. Write a program that will ask the user for a value of pitch and a value of duration and then produce that tone.

2. Using READ/DATA statements, have the computer play a variety of notes. Develop some interesting sound effects for your programming library.

Topic 17
SOUND GENERATION—APPLE—PART II

Performance Objectives: Students will be able to:

- poke a machine language sound routine into computer memory providing musical tone capability
- learn to access and use the installed machine language routine from a BASIC program
- understand the effects of PITCH and DURATION upon musical tones
- learn to use sound routines to "spice up" CAI programs or any BASIC program

Vocabulary: New: pitch, duration, rest, machine language
Review: POKE, PEEK, GOSUB, FOR/NEXT

Motivation:

- Describe at least three ways musical tones enhance and improve programs such video arcade games.

- Describe at least three ways musical tones enhance and improve educational programs such as CAI (Computer Assisted Instructional) programs.

Aim: How can we use sound to enhance and add interest to BASIC programs?

Development:

1. Programmers like to use machine language routines because they are processed at a higher speed than BASIC program statements. Why do you think this is so? (Because machine language commands do not have to go through an interpreter like BASIC statements.)

2. With a machine language routine we can drive the speaker at a higher frequency (cycles per second) and therefore produce higher pitched tones.

3. Remember that the faster we can pulse or cycle the speaker the higher the pitch will be, and that the pitch of the tone produced will depend on the relative number of times (frequency) we pulse the speaker.

4. The length of time a particular tone is played also affects the quality of that note. This is called the duration. The greater the duration the more intense that note will be.

5. What else may affect the tones produced by the computer?

   Acceptable answer:

   In addition to pitch and duration, a third element called "rest" will alter musical notes. Rest is the amount of time or pause between one note and the next. To program this in BASIC we can use the FOR/NEXT loop as a timing loop:

```
501 R = 100
502 FOR X = 1 TO R: NEXT
```

6. How may we install a machine language sound routine?

Acceptable answer:

One way to place machine language instructions into the memory of the computer is to POKE the decimal values of the codes into an appropriate area of memory.

- Many programmers prefer to use the memory area located between 768 and 1023 ($300 - $3FF in hexidecimal) for short machine language routines.

- The following subroutine is used to install the machine language routine necessary to drive the speaker on the Apple II, II+, IIe computer:

```
5000 POKE 770,173: POKE 771,48:  POKE 772,192: POKE 773,136
5010 POKE 774,208: POKE 775,5:   POKE 776,206: POKE 777,1
5020 POKE 778,3:   POKE 779,240: POKE 780,9:   POKE 781,202
5030 POKE 782,208: POKE 783,245: POKE 784,174: POKE 785,0
5040 POKE 786,3:   POKE 787,76:  POKE 788,2:   POKE 789,3
5050 POKE 790,96
5060 RETURN
```

7. Once the above has been implemented with a GOSUB 5000, we may then consider the sound routine as being "installed" into the computer's memory. As long as the computer is not turned off, this routine can be accessed by any program which correctly uses the "installed" routine.

8. How can we program our computer produce tones once the sound routine has been installed? (Any integer value from 1 to 255 may be used for the P, D and R variables which are the pitch and duration and rest values respectively.)

Example:

```
10 GOSUB 5000: REM Install sound routine
20 R=100:P= 100:D=75:GOSUB 500
30 R=50 :P= 200:D=90:GOSUB 500
40 R=75 :P= 150:D=80:GOSUB 500
50 END
500 REM Play a note
502  FOR X = 1 TO R: NEXT
504  POKE 768, P: REM  PITCH
508  POKE 769,D: REM Note length
510 CALL 770: REM Play the note
512  RETURN
```

Applications:

1. How can we produce the musical scale?

The following routine will produce a musical scale:

```
101 D=80 : R=80
102 P = 63:  GOSUB 500: REM C
103 P = 84:  GOSUB 500: REM D
104 P = 103: GOSUB 500: REM E
105 P = 111: GOSUB 500: REM F
106 P = 127: GOSUB 500: REM G
107 P = 141: GOSUB 500: REM A
108 P = 153: GOSUB 500: REM B
109 P = 159: GOSUB 500: REM C
```

2. In order to observe the effect of pitch upon the tone, create a program so that the duration remains 100 (D=100) but the pitch (P) is increased by fives from 5 to 255. What is the effect upon the tone by increasing the value of P?

3. In order to observe the effect of duration upon the tone, modify the program above so that the pitch remains 100 (P = 100) but increase the value of D by fives from 5 to 255. What is the effect upon the tone by increasing the value of D?

4. In order to observe the effect of rest upon the tone, modify the program above so that the pitch remains 100 (P = 100) and the duration remains 100 (D= 100), but increase the value of R by fives from 5 to 255. What is the effect upon the tone by increasing the value of R?

Summary:

1. Why do we need a machine language routine for sound?

2. How can we enter a machine language program from BASIC?

3. How can we produce desired tones once a driver routine is installed?

4. What are the effects of Pitch, Duration and Rest upon musical tones?

Assignments:

1. Using a FOR/NEXT loop, create a subroutine which will play tones from 1 to 255 with a constant duration of 10.

2. Write a program that will produce 10 tones increasing in pitch.

3. Write a program that will produce the same note 10 times increasing in duration.

4. Write a program that will produce three notes increasing in pitch followed by three notes decreasing in pitch all with different durations.

Assignments for More Advanced Students:

1. Write a program that will ask the user for a number from 1 to 100, and then with the random generator function of the computer, produce that many random tones. Each tone should have a different P, D and R value.

2. Using the musical scale as a guide, compose a simple but familiar melody.

Topic 18
INTRODUCTION TO APPLE SEQUENTIAL TEXT FILES

Performance Objectives: Students will be able to:

- understand when text files are needed and appropriate
- describe the physical characteristics of a sequential text file
- write a program to input a sequential text file

Vocabulary: New: OPEN, READ, CLOSE, Sequential Text File
Review: DOS, CTRL-D

Motivation:

- We know that we can save programs to the disk and recall them for future use. Certain programs, however, can take information and save the information, as a text file, to the disk in addition to the program itself. Thus the information (stored as a text file) can be loaded by the originating program, or another program, in order to use the text file. Life becomes a lot easier by not having to re-enter previous information.

- Discuss how the loading of previously saved text files is important to the process in each instance below:

  - Editing yesterday's written homework assignment on a word processor.

  - You just sold two more auto parts (#3402) and wish to change your computer inventory accordingly.

  - In a modern library you wish to see if the book you wish to take out is in the library.

  - You wish to dial up an electronic bulletin board service in order to look up the history of a company in which you wish to invest.

  - Can you think of additional examples?

Aim: How can we read stored information from a sequential text file into the memory of the computer?

Development:

1. What kind of information can be stored on a disk? (both text and numbers)

2. What kinds of files are there? (Apple DOS (Disk Operating System) supports two types of disk files: sequential text files and random access files. The latter will be discussed in another lesson.)

3. What is a Sequential Text File? (It consists of a number of records separated by carriage returns and stored one after the other on a disk or tape.)

4. How can you tell if there is a text file on disk?

Answer:

When you catalog an Apple disk the letter "T" will indicate that it is a text file:

T 002 ANIMALS

5. Sequential text files, as indicated by the name, are read into the computer's memory in successive manner, one character at a time.

   . Where CR stands for carriage return, the data will look like this:

   item 1 CR item 2 CR item 3 CR item 4 CR

   . One of the characteristics of a sequential text file is that in order to retrieve one of the pieces of information, it is necessary to read all of those which precede it.

6. Can you think of examples of when you might wish to read all the file into computer memory?

   . word processing
   . text file is relatively short
   . notes of a song

7. How can we read a sequential text file into memory? (We must learn to use the following DOS commands: OPEN, CLOSE, READ.)

   . The following file will read in the five animal names:

```
10 REM ANIMAL READ PROGRAM
20 D$= CHR$(4): REM DOS CONTROL STRING
30 PRINT D$ "OPEN ANIMALS"
40 PRINT D$ "READ ANIMALS"
50 FOR I = 1 TO 5
60 INPUT A$(I)
70 NEXT I
80 PRINT D$ "CLOSE ANIMALS"
90 END
```

   . Note that the INPUT statement is used to "extract" each piece of data from the file. Also each piece of data is assigned an appropriate variable so that it can be stored in the computer's memory. Thus, to print animal #3 we simply PRINT A$(3).

Applications:

1. How would you modify the ANIMAL READ program to print out all the animals, after they have been read into the computer's memory?

   An acceptable answer is to add the following:

```
90 FOR I= 1 TO 5
100 PRINT A$(I)
110 NEXT I
```

2. Not all of the file has to be read into memory, but it must be read in sequentially. Thus, how could you read in only the first three elements of the ANIMALS file? (Change the FOR/NEXT loop to read FOR I = 1 to 3.)

3. How could we read in a file that has two array elements such as a person's name and age?

```
10 REM NAME AND AGE
20 D$= CHR$(4): REM DOS CONTROL STRING
30 PRINT D$ "OPEN FILENAME"
40 PRINT D$ "READ FILENAME"
50 FOR I = 1 TO 5
60 INPUT N$(I)
65 INPUT A$(I)
70 NEXT I
80 PRINT D$ "CLOSE FILENAME"
90 END
```

- Note we could combine the INPUT line with the comma since the comma is used as a data field separator.

```
60 INPUT N$(I), A$(I)
```

Summary:

1. How are text files used to save time and effort?

2. What is a sequential text file?

3. How can we read a sequential text file on an Apple computer?

4. How can we store into the computer's memory the information obtained from a text file?

Assignments:

1. Write a program that will read 15 names of students and print them on the screen in three columns and five rows.

2. Write a program that will read in ten students and their test scores. Compute the class average, print each student and whether or not that student is above or below average.

3. Write a program that will read in 100 numbers and print those numbers which are greater than 50.

4. Write a program that will read in 10 students and their last three test scores.

Assignments for More Advanced Students:

1. Write a program that will read in 10 students and their scores and then print them in order of highest to lowest score.

2. Add to the above program a second printing of the names and scores in

alphabetical order.

3. Create a program which will store names in a text file.

A sample program to Create a Text File With Five Animal Names

```
 10 REM ANIMAL WRITE  PROGRAM
 20 D$= CHR$(4): REM DOS CONTROL STRING
 30 PRINT D$ "OPEN ANIMALS"
 40 PRINT D$ "WRITE ANIMALS"
 50 PRINT "TIGER"
 60 PRINT "LION"
 70 PRINT "BEAR"
 80 PRINT "CAT"
 90 PRINT "DOG"
100 PRINT D$ "CLOSE ANIMALS"
110 END
```

Topic 19

COMMODORE TAPE AND DISK SEQUENTIAL INPUT COMMANDS

Performance Objectives: The students will be able to:

- describe the purpose of a sequential file
- write input to a Commodore sequential tape file
- write input to a Commodore sequential disk file

Vocabulary: New: sequential file, offline, online

Motivation:

- If you wanted to sort thousands of names, how would you store the information in your computer? (A sequential data file on a tape or disk will allow the user to store huge amounts of data.)

Aim: How do you read data from a Commodore sequential data file?

Development:

1. A file is a collection of information which can be stored on a tape or a diskette. Files allow the user to have access to huge quantities of data offline, that is not in the computer. Storage offline is less expensive than online, that is computer storage. It takes more time, though, to access data offline than online. In a sequential data file, information is stored and retrieved in order, that is, sequentially. Another type of data file, random access data files, allow the user to more readily retrieve information in the middle of a file. However, random access files require more storage area.

2. A sequential file first needs to be opened. Then the data is read. Finally, the file must be closed. The commands are different for a tape and a diskette file.

3. The OPEN command for a tape is:

   OPEN N,D,S,FILENAME

   The N is an integer which refers to the logical file number. Usually, the first file you open would have N=1.

   The D refers to the cassette number, whether tape 1 or tape 2. The default condition for D is 1.

   The S refers to the operation that is to be performed. Read is indicated by a 0 and write is indicated by a 1. The default condition is 0. If the filename is not indicated, the first file is accessed. Filenames must have 16 characters or less.

4. The read command for a tape is either Input#N (for reading numeric and string fields from a data file) or Get#N (for reading one character at a time). Note that the N is the number used in the open statement.

5. The close command is:
   CLOSE N

6. An example of reading 20 numbers from a tape file is:

```
10 OPEN 1,1,0 "DATAFILE"
20 FOR I=1 TO 20
30     INPUT#1,I
40     PRINT I
50 NEXT I
60 CLOSE 1
```

Note that step 40 prints the numbers on the screen.

7. An example of reading 20 letters from a tape file is:

```
10 OPEN 1,1,0,"DATAFILE"
20 FOR I=1 TO 20
30     GET#1,I
40     PRINT I
50 NEXT I
60 CLOSE 1
```

8. You open diskette files with the DOPEN# statement (for BASIC 4.0 on the Pet). The form is:

DOPEN#N,FILENAME,D,W

to open the file. The filename must be included. D refers to D1, drive 1. The default condition is D0, drive 0. The W refers to the write operation. The default condition is a read operation. An example of an open statement is:

DOPEN#1,"DATAFILE",D1. This opens file "DATAFILE" for a read operation on drive 1.

9. The close statement is:

DCLOSE#N

If a file is not closed, some data may be lost.

10. Note that you cannot open a new sequential file for a read operation. The file must already have been written to, or otherwise you will get a FILE NOT FOUND ERROR.

Applications:

1. Write a program to read the names of 100 customers from a diskette data file.

2. Write a program to read 200 characters from a diskette data file.

3. Write a program to read 100 numbers from a cassette data file.

Summary:

1. What is a sequential data file?

BEST COPY AVAILABLE

2. What is the open statement for cassette data files?

3. What is the difference between an input and a get statement?

4. What is the open statement for diskette data files?

Assignments:

1. Write a program to read 100 names, addresses, and numbers from a cassette data file.

2. Write a program to read 500 student's names, ID numbers, and grades from a diskette data file, and print the average grade for each student's data file. Print the highest and lowest grades.

Assignments for More Advanced Students:

1. Write a program to read 20 letters from a diskette data file. Print the letters in alphabetical order.

2. Read 500 student's names, ID numbers, and marks from a diskette data file. For one list, print the marks with the corresponding names and ID numbers in ascending order. For another list, print the names, with their corresponding ID numbers and marks, in alphabetical order.

Topic 20

IBM PC AND TRS-80 DISK SEQUENTIAL FILES

Performance Objectives: Students will be able to:

- open a disk sequential data file using one of three modes of access
- to transfer data from memory to disk using PRINT#1 statement
- to transfer data from disk to memory using INPUT#1
- to close a disk sequential file
- to create a sequential file
- to add records to an existing file
- to eliminate a file
- to delete a record from a file
- to retrieve a record from a file
- to update a file

Vocabulary: New: PRINT#1, INPUT#1, LINE INPUT, OPEN , CLOSE, "O", "E", "I", EOF
Review: record, file, sequential file, PRINT#-1, INPUT#-1

Motivation: What is the big advantage to using a disk over using a cassette as your auxiliary memory device? (speed)

Why would that become important if we were trying to use a file of all the names and information about each person in our school in order to:

1) Creating pupils' programs
2) Recording students grades and generating report cards
3) Maintaining permanent records
4) Monitoring and maintaining attendance records
5) Generating reports and lists: pupils available for bus passes, honor roll, pupils with siblings within school, students who are candidates for graduation, eligible for Arista or Honor Society, etc.

Acceptable answer: We would save a lot of time

Aim: How do we create and use disk sequential data files?

Development:

1. When you access BASIC your machine asks the question:
   "HOW MANY FILES?-"

   You can type any number between 1 - 15 to create 1 - 15 buffers to handle the number of files you will use. If you just press ENTER , the default number of buffers created is three. A buffer is an area of RAM where datum is stored temporarily when it is going to and from a disk file.

2. With manual files, all records are kept in file cabinets and in order to use the files, a person must open the drawer or folder which contains the file. When finished with the file, the drawer should be closed.

   Disk files also must be OPENed and CLOSEd in order to be accessed.

3. The OPEN statement tells the computer which buffer to use and also what type of access you want.
   There are three types of access:

   (a) "O" Sequential Output. Data transferred from memory to disk. This creates a file if it doesn't already exist and writes the new data OVER the existing data.

   (b) "I" Sequential Input. Data transferred from file to computer memory starting at the beginning and proceding in order to the end. Prints FILE NOT FOUND if the file does not exist.

   (c) "E" Sequential Output at End of File. Data transferred from memory to disk. This creates a file if it doesn't already exist BUT writes the new data at the END of the file, not destroying previously stored data.

   (d) "R" Random Input and Output. This will be discussed in another lesson.

4. The format for the OPEN statement is:

   OPEN"R",1,"FILENAME"

5. If a file is OPENed, it must be closed. The format is:
   CLOSE buffer or channel number

6. Once a file is OPENed, you can process the information by using the PRINT# and INPUT# commands similar to cassette sequential files except instead of -1 as the channel for cassette access, you type the number of the buffer you are using for that file.

   (a) PRINT# channel, variable
   (b) If you have a combination of variables that will be transferred to disk at the same time insert commas between quotation marks to act as markers to differentiate data items.

   i.e. PRINT# channel, variable; ","; variable
   (c) If you have inputted data that has embedded commas you must use the notation:

   LINE INPUT string variable

   before your INPUT#1 statement.

7. Write a program module that will save a list of names on a disk. Use "LAST NAME" to indicate the final record.

<u>Acceptable answer:</u>

```
10 REM*** This will create a file on a disk        ***
20 REM***  of a list of names                       ***
30 CLS                    :REM  clears screen
40 OPEN "O", 1, "NAMES:1"        :REM the ':1' means disk drive #1
50 PRINT  "Type in the name to be stored"
```

```
 60 PRINT "Type 'LAST NAME' when you are finished"
 70 INPUT N$              :REM keyboard to memory
 80 IF N$ = "LAST NAME" THEN 999     :REM  exit
 90 PRINT#1, N$           :REM memory to cassette tape
100 GOTO 50
110 CLOSE 1
999 END
```

8. The statement to transfer information from disk to memory is INPUT#1.

9. Write a program that will retrieve a file of names stored on disk and print each record on the screen

Acceptable answer:

```
 10 REM*** This program will transfer information ***
 20 REM***  stored on a disk into the              ***
 30 REM***  computer's memory and then display     ***
 40 REM***  record on screen                       ***
 50 OPEN "I", 1, "NAMES:1"
 60 CLS                        :REM clears screen
 70 INPUT#1, N$                :REM disk to memory
 80 IF N$ = "LAST NAME" THEN 999  :REM exit
 90 PRINT N$                  :REM  memory to screen
100 PRINT
110 GOTO 70
120 CLOSE 1
999 END
```

10. The EOF (End of File) detector has the format:

    EOF(channel)

    which returns a logical true (-1) when the end of the file has been reached; otherwise, a logical false (0) is returned.

    ex.
```
100 IF EOF(3) THEN PRINT "END OF FILE"
100 IF EOF(2) THEN CLOSE 2
```

An EOF and a GOTO loops are used to read most sequential files

Applications:

1. Write a program that will store 5 additional INPUTed pupils' names and averages on a disk sequential file called "HOMEROOM"

Acceptable answer:

```
 10 REM***  This program will add 5 names and avs. ***
 20 REM***  to an existing sequential disk file    ***
 30 REM***                                        ***
 40 REM*************************************************
 50 OPEN "E", 1, "HOMEROOM:1"
 60 FOR X = 1 TO 5
 70     PRINT "TYPE IN THE NAME TO BE ADDED"
 80     INPUT N$
```

```
 90     PRINT "TYPE IN THE AVERAGE"
100     INPUT AV
110     PRINT#1, N$;",";  AV
120     PRINT
130 NEXT X
140 CLOSE 1
999 END
```

2. Write a program that can be used to build a series of files that contain form letters to be used by the P.T.A. of your school.

Acceptable Answer:

```
 10 REM***    This program allows you to take a letter  ***
 20 REM***    and place it in a file                    ***
 30 REM***        N$ = Name of letter                   ***
 40 REM***        N = Number of lines                   ***
 50 REM***        L$ = Individual lines                 ***
 60 REM***                                              ***
 70 REM*********************************************************
 80 CLEAR 1000
 90 PRINT "Type in the name of the form letter you wish to use."
100 INPUT N$
110 LET N$ = N$+":1"
120 OPEN "O", 1, L$
130 PRINT "What is the total number of lines in the letter?"
140 PRINT "  Be sure to count blank lines in your total."
150 INPUT N
160 PRINT "Type in your letter one line at a time."
170 FOR L = 1 TO N
180    LINE INPUT L$        :REM allows commas
190    PRINT#1, L$
200 NEXT L
210 CLOSE 1
999 END
```

Summary: What keywords are used to open or close a sequential file?
How can a sequential file be created?
How many fields can a single record have?
Can we add to or delete from a sequential file?

Assignments:

1. Write a program which will allow a user to store 35 names, addresses, phone numbers and grades (five for each student). The program should allow for searches by name.

2. To program #1 above, add a module which will allow additions to the file. The program should also allow a printer to be used for searches.

3. To program #1 above, add the facility for the user to search for any first name or grade above or below a given amount.

Topic 21
MANAGING APPLE TEXT FILES

Performance Objectives: Students will be able to:

- write a program that will create a text file
- understand how text files are used to store information
- create and read text files into an array
- learn to input a text file, make changes and output the new information to the disk as a text file

Vocabulary: New: WRITE
Review: OPEN, READ, CLOSE, INPUT, PRINT, sequential text file, array, FOR/NEXT loop

Motivation:

- Pretend that you have been hired by the Safari Computer Company to create a new arcade game called POKER POWER. As part of your program you will need to create text files representing playing cards.

- Describe in words the type (categories) of information you might want to store in the text file in order to describe the playing card. Elicit: number, shape (club, diamond, heart or spade), color, picture card, etc.

Aim: How can we output a sequential text file to disk for permanent storage?

Development:

1. How can we write a program that will store the features of a playing card?

   An acceptable solution might be as follows:

   We decide to store three pieces of information for each card. The first is the value of the card represented by A, K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3, 2. The second is the value S, C, D, H, representing a spade, club, diamond and heart respectively. The third is R or B for the color red or black.

   Next we will create a program to store 5 cards onto disk:

```
10 REM Playing Card Write PROGRAM
20 D$= CHR$(4): REM DOS CONTROL STRING
30 PRINT D$ "OPEN CARDS"
40 PRINT D$ "WRITE CARDS"
50 PRINT "K,S,B"
55 PRINT "9,D,R"
60 PRINT "2,C,B"
65 PRINT "5,C,B"
70 PRINT "K,H,R"
80 PRINT D$ "CLOSE CARDS"
90 END
```

   RUN this program with an initialized disk in the drive and you will see

the "in use" light come on. You have stored your information as a text file.

2. What command causes the computer to output information? (The "WRITE" command followed by "PRINT" statements.)

   (Note: The data written to disk must be in quotation marks. The comma is to separate one piece of data from the next.)

3. Can you create a program that will retrieve the information you have stored?

   An acceptable solution might be as follows:

```
10 REM PLAYING CARD READ PROGRAM
20 D$= CHR$(4): REM DOS CONTROL STRING
30 PRINT D$ "OPEN CARDS"
40 PRINT D$ "READ CARDS"
50 FOR I = 1 TO 5
55 INPUT V$(I)
60 INPUT S$(I)
70 INPUT C$(I)
80 PRINT D$ "CLOSE CARDS"
90 END
```

4. Discuss how data is stored in arrays. In our example, the V$ is the string array for the value of the card, and S$ and C$ are the string arrays for storing the suit and color respectively.

5. How would you print the description of the fourth card?

   (100 PRINT V$(4), S$(4), C$(4))

Applications:

1. After reading the five cards into the memory of the computer, how would you program the computer to inform the user of the color of the card?

   An acceptable solution might be as follows:

```
200 IF C$(4) = "R" THEN PRINT "THIS CARD IS RED"
```

2. What steps would you take to change an existing card?

   . Load CARD text file into memory (see Playing Card Read Program).

   . Add following program statements:

```
200 PRINT "WHICH CARD NUMBER TO CHANGE";:INPUT N
205 PRINT "THE CARD IS A ";: INPUT V$(N)
210 PRINT "WHAT IS THE SUIT? (C,S,D,H)";:INPUT S$(N)
215 PRINT "WHAT IS THE COLOR? (R,B);:INPUT C$(N)
```

   . SAVE CARD text file (see Playing Card Write Program).

3. Describe applications where it is necessary to recall information, make

changes and save the new information.

Summary:

1. How do we write a sequential text file?

2. How do we read a sequential text file?

3. How can the information from a text file be stored in the computer's memory?

4. What are the procedures necessary to edit a text file?

Assignments:

1. Write a program that will store the names of five students and their test scores as a text file.

2. Write a program that will ask you for three sentences (INPUT statements). After the third sentence has been entered, it will store them as a text file.

3. Describe three additional situations (not discussed in class) where the ability to load, edit and save text files is useful.

4. Using FOR/NEXT loops to save work, write a program that will write all the cards of a playing deck to the disk.

Assignments for More Advanced Students:

1. Modify the Playing Card Read Program so that it will read five cards, but also will count and print the number of red and the number of black cards.

2. Modify the Playing Card Read Program so that after reading the five cards it will ask the user to guess any of the cards. Give the user three chances to guess any of the cards in the computer's memory.

Topic 22
APPLE RANDOM TEXT FILES

Performance Objectives: Students will be able to:

- understand the difference between sequential and random text files
- write a program that will create a random text file
- understand when random text files are needed and appropriate
- write a program to recall a specific portion of a random text file

Vocabulary: New: random access text file, "R" parameter, "L" parameter
Review: DOS, CTRL-D, open, read, write, close, sequential text file

Motivation:

- A random access file can be thought of as an encyclopedia with many volumes. If you want to find a piece of information, you can just pull out the appropriate book without having to go through the ones that come before it. Unlike sequential files, random files allow you to look at any part of the file, without having to look at the material in the beginning, middle or end.

- Many applications in the physical world require a random application rather than a sequential one. For example, it would seem quite silly to look up a friend's phone number by turning through every page in your phone directory until you found the correct one.

  - Can you describe three other normally random applications, which become silly when a sequential approach is applied?

Aim: How can we output and input random access text files to disk for permanent storage?

Development:

1. What is a random access text file? (It is a text file very similar to the sequential text file we discussed in an earlier lesson. Random text files are used to store and retrieve information on disk almost the same way sequential files are stored. That is, programs must be created in order to read or write these files.)

2. How does a random access text file differ from a sequential text file? (The same commands used to create a sequential text file are used to create a random access text file with the addition of an "L" or length parameter in the name and an "R" or record parameter when reading or writing records.)

3. Can we write a random access program to store names and phone numbers?

Answer:

Enter the following program:

```
10 REM PHONE DIRECTORY WRITE PROGRAM
20 D$= CHR$(4): REM DOS CONTROL STRING
30 NAME$(1) = "ADAM L. 456-3498"
40 NAME$(2) = "BAKER J. 516-223-4321"
50 NAME$(3) = "SMITH A. 374-4848"
60 PRINT D$;"OPEN PHONE DIRECTORY,L40"
70 PRINT D$;"WRITE PHONE DIRECTORY,R1"
80 PRINT NAME$(1)
90 PRINT D$;"WRITE PHONE DIRECTORY,R2"
100 PRINT NAME$(2)
110 PRINT D$;"WRITE PHONE DIRECTORY,R3"
120 PRINT NAME$(3)
130 PRINT D$;"CLOSE PHONE DIRECTORY"
140 END
```

- Since no record will require more than 40 characters, we selected 40 as our length parameter.

- Note that each time we write to a random access text file, we must use the "R" (record number) parameter.

- RUN this program with an initialized disk in the drive and you will see the "in use" light come on. You have stored your information as a random access text file.

4. How can we read the Phone Directory text file?

```
10 REM PHONE DIRECTORY READ PROGRAM
20 D$= CHR$(4): REM DOS CONTROL STRING
30 PRINT D$; "OPEN PHONE DIRECTORY"
40 FOR I = 1 TO 3
50 PRINT D$;"READ PHONE DIRECTORY,R";I
60 INPUT NAME$(I)
70 NEXT I
80 PRINT D$;"CLOSE PHONE DIRECTORY"
```

Applications:

1. Can we retrieve only the second record of our random file?

Answer:

Try the following program:

```
10 REM READ SECOND RECORD ONLY
20 D$= CHR$(4): REM DOS CONTROL STRING
30 PRINT D$; "OPEN PHONE DIRECTORY"
50 PRINT D$;"READ PHONE DIRECTORY,R2"
60 INPUT NAME$
70 PRINT D$;"CLOSE PHONE DIRECTORY"
80 HOME: PRINT NAME$
```

2. Describe the procedure to change someone's phone number on our Phone Directory file. (Open file, read in record, input new phone number, write record, close file.)

3. How could you prevent the user from inputting new information which will exceed our record limit of 40 characters?

   Answer:

   Error trap the input statement:

```
100 PRINT " NEW NAME AND NUMBER? ";:INPUT N$
105 IF LEN( N$ )>40 THEN 9999
9999 PRINT "RECORD LENGTH EXCEEDED": GOTO 100
```

4. How can we modify our Phone Directory Write Program so that the name and phone are separate fields in the same record?

```
30 NAME$(1) = "ADAM L."
35 NUMBER$(1)= "456-3498"
40 NAME$(2) = "BAKER J."
45 NUMBER$(2)="516-223-4321"
60 PRINT D$;"OPEN PHONE DIRECTORY,L40"
70 PRINT D$;"WRITE PHONE DIRECTORY,R1"
80 PRINT NAME$(1); NUMBER$(1)
90 PRINT D$;"WRITE PHONE DIRECTORY,R2"
100 PRINT NAME$(2); NUMBER$(2)
130 PRINT D$;"CLOSE PHONE DIRECTORY"
140 END
```

Summary:

1. How do we write a random text file?

2. What advantages does a random access file have over a sequential text file?

3. How can we change a specific record in a random file?

4. What are the procedures necessary to edit a random access text file?

Assignments:

1. Write a program that will store five students and their test scores as a random access text file.

2. Write a program that will ask you to select which student score you wish to see, and pull out and print that record from the random access file created by the above assignment.

3. Write a program that will create a random access text file of at least five people. Each record should be a maximum of 75 characters, containing a separate field for first name, last name, address and phone number.

Assignmments for More Advanced Students:

1. Write a program that will ask the user which record to see, and have it print that record on the screen. Also, ask if the user wishes to change that record. If so, INPUT and WRITE the new information to disk.

2. Write a database program that stores student names and grades (separate fields) as a random access text file. Include the option to change any record and print the average test score.

Topic 23
IBM PC AND TRS-80 RANDOM ACCESS FILES

Performance Objectives: Students will be able to:

- identify direct (or random) access files
- compare and contrast sequential access files and direct (or random) access files
- create direct (random) access files
- access direct access files

Vocabulary: New: direct access files, random access files, LSET, RSET, CVI, CVS, CVD
Review: FIELD, disk file, GET, PUT, MKI$, CLOSE, LOC, buffer

Motivation:

- Data that has been accumulated may be stored outside of the machine. How may this be done? (Data may be stored in a disk file.)

- What is a disk file? (A disk file is an organized grouping of related data.)

- If data is accessed in the same order it has been stored, what type of access is this? (sequential access)

- What are sequential access files? (Sequential access files are files that may be accessed only in the order the data has been stored, i.e. sequentially.)

- Is there a way of accessing the data "out of order," as needed or must all data files be stored sequentially?

Aim: How may data be stored so that it may be accessed as needed in a "random" order?

Development:

1. How is a sequential access file created? (Give an example of a programming command to open or create the file.) (OPEN "O", 1, "LIST/ADD")

   - What does this mean? (The file is opened in "OUTPUT" mode. It is assigned the buffer number 1 and is named LIST/ADD.) (Note: A number from 1 to 15 may be assigned as a buffer number.)

2. A direct access file may be created in a similar manner.

   - To create a direct access file the file is again opened but in direct access mode as follows: OPEN, "D", 1, "LIST",24

   - How are the direct access and sequential access statements similar? (OPEN is used as the command, the buffer number is assigned, and the file is named.)

- What does the "D" represent in the direct access file command? (The D represents the direct access mode used to create the direct access file.) (Note: The letter "R" may be used interchangeably with the letter "D" in this command as the "R" stands for random access, another name for direct access.)

- Why is the number 24 included within the programming statement? (It is used to set the record length to 24 bytes.) (Note: If the record length is not specified within the OPEN statement, the computer will assume it to be 256 bytes.)

3. What is the advantage of a direct access file? (Data can be accessed from almost anywhere on the disk. It is not necessary to read through all data sequentially to proceed to a specific record.)

   - A "record" is a distinct unit of information. Each record is numbered.

     (Note: BASIC assigns space for records in numeric order. Thus, if the first record written to the file is number 125, BASIC allocates space for records 0 through 124 before record 125 is stored in the file. The maximum number of logical records is 65,535. Each record may contain up to 256 bytes.)

4. How is space allocated in the buffer for the variables that are written to a direct access file? (Use a FIELD statement.)

   - What statement would be used to allocate the first 30 positions in buffer 1 to string K$? (FIELD 1, 30 AS K$)

   - What would the following statement represent? FIELD 1, 20 AS A$, 14 AS B$, 12 AS C$ (The statement would allocate the first 20 positions in buffer 1 to A$, the next 14 to B$ and the next 12 to C$.)

5. Once the buffer is set up, data may then be moved into it using the LSET statement.

   - What type of variables are placed within a buffer? (string variables)

   - How may numeric variables be placed within a buffer?

   - Review the "make" functions which convert numeric variables into a string, using MKI$.

   - If we wish to INPUT the names and ages of a class into a program and store the data within a file, how may we move the data into the buffer?

     Sample program:

```
70 INPUT "NAME"; N$
80 INPUT"AGE"; A
90 LSET A$=N$
100 LSET B$=MKI$(A)
```

6. Once the data is within the buffer, what might we want to do with this data?

- After data is within the buffer, it may be written to a record within the direct access file. This may be accomplished by using the PUT statement, indicating the buffer from which the data is coming followed by the name of the record it is being written to, e.g., PUT 1, NAME%

  (Note: The percentage sign at the end of a variable specifies that the variable is an integer.)

7. How can you be sure that all records have been written to the disk? What programming statement will do this for you? (CLOSE)

   - The CLOSE statement will close access to a file. To close a specific buffer, the number of the buffer must be indicated within the statement, such as CLOSE 2, CLOSE 15, etc. If the buffer is not indicated within the CLOSE statement, all open files will be closed.

8. Type in the following program and determine what occurs. Identify what occurs within each line of the program.

   Program:

```
10 OPEN"D", 1, "LIST", 32
20 FIELD 1, 12 AS F$, 20 AS L$
30 INPUT"INPUT 0 TO END"; CODE%
40 IF CODE%=0 GOTO 110
50 INPUT"FIRST NAME";F$
60 INPUT"LAST NAME"; L$
70 LSET A$=F$
80 LSET B$=L$
90 PUT 1, CODE%
100 GOTO 30
110 CLOSE
```

   (Note: Program will permit the user to store the first name and last name of persons within a disk file.)

9. Now that the data is stored within the disk, how might we proceed to eventually get this data out?

   - We again must OPEN the file in "D" mode, then we must allocate space within the buffer for the variables that will be read from the file.

   - Next, we must GET the record from the direct disk file and place it within the buffer. The GET statement will do this task, just as the PUT statement originally placed the record within the file, e.g., GET 1, NAME%

10. What command was used to rename numeric variables as string variables when data was moved into the buffer? (MKI$(X))

    - How are the variables now stored within the buffer? (as string variables.)

    - When these variables are later accessed, must they be accessed as string variables?

- The CV statement will convert string values back to numbers, using the convert function, according to the type of numeric variable that was originally made into a string variable.

  CVI(A$) converts a string variable to an integer
  CVS(A$) converts a string variable to single-precision values
  CVD(A$) converts a string variable to double-precision values

11. Write the program which will access the direct-access file list which was created in the sample program above.

```
10 OPEN"D", 1, "LIST", 32
20 FIELD 1, 12 AS F$, 20 AS L$
30 INPUT "TYPE IN 0 TO END THE FILE";CODE%
40 IF CODE% = 0 THEN 90
50 GET #1, CODE%
60 PRINT B$
70 PRINT A$
80 GOTO 30
90 CLOSE 1
```

Summary:

1. How are sequential access files and direct access files similiar?

2. What is the advantage of a direct access file?

3. What is the buffer used for?

Assignmments:

1. Draw a chart tracing the steps taken to create a direct access file and the steps taken to access a direct access file.

2. Create a random access file made up of the following information in the order it is written:

   First name, last name, age, phone number

   After the file is created, access the file and print out the information sequentially by age in the following order:

   Last name, first name, phone number, age

3. Write a program which will store 20 random numbers each less than 1000, within a direct access file.

4. Create a poster illustrating the mechanics of a direct access file and the mechanics of a sequential access file.

Topic 24
TRS-80 CASSETTE FILES

Performance Objectives: Students will be able to:

- state the difference between a data record and a data file
- state three advantages in using data files
- to transfer data from memory to cassette using PRINT#-1 statement
- to create a cassette file

Vocabulary: New: record, file, sequential file, INPUT#-1, PRINT#-1
Review: string, numeric, I/O devices

Motivation:

- What are some of the ways that our school uses (could use) computers for student's names, homerooms and other data about pupils?

  Acceptable answer:

  1) Creating pupils' programs
  2) Recording students' grades and generating report cards
  3) Maintaining permanent records
  4) Monitoring and maintaining attendance records
  5) Generating reports and lists: pupils available for bus passes, honor roll, pupils with siblings within school, students who are candidates for graduation, eligible for Arista or Honor Society, etc.

- What would our biggest problem be if we were to write a program that would compute the average of each pupil's grades for last term?

  Acceptable answer:

  Entering the name and grades for each pupil in the school

- Data can be entered and saved on cassettes and disks so that it can be used with many programs. These lists of related items are called data files. Individual entries on a data file are called "records."

Aim: How do we use cassette sequential data files?

Development:

1. Review the ways previously used in class of entering lists of data which and discuss the drawbacks of each.

   Acceptable answer:

   INPUT: If your program is used again data must be entered again.

   READ - DATA: If new datum is necessary, the program must be rewritten each time and the new data must replace the old data.

   Data has to be retyped for each program that needs to use it.

2. What are some lists that might be kept in a personal computer?

Acceptable answer:

A) telephone numbers of friends
B) addresses of friends
C) birthdays and anniversaries
D) inventory of hobby materials (coins, stamps, comics, etc.)
E) inventory of personal property with serial numbers and other purchase information (T.V., computer and peripherals, kitchen appliances)

3. Each of the above lists would be a data file. Data sets or files are stored on auxiliary memory devices such as cassettes, disks, or punched cards.

- Each entry within the list is called a data record. For instance, in your telephone number file, you might have a record such as "Paul Bodin 273-3125." Another record in the telephone file might be "Janet Johnsen 727-5213." Each record could have many other items, such as address, zipcode, birthday, etc.

- SEQUENTIAL files are files that can be accessed only from the very beginning to the file whenever the computer is searching for data. The computer processes the data in order (sequence) from the beginning to the end or stopping record.

4. Creating cassette data files:

- Review purpose of these two lines of code:

```
10 INPUT A$
20 PRINT A$
```

Acceptable answer:
line 10 transfers characters from keyboard to a memory location it designates as A$
Line 20 transfers characters stored in memory location,A$, and prints them on CRT

- to transfer a character stored in location A$ to the cassette player instead of the screen, line 20 has to be modified to read:

```
20 PRINT#-1, A$
```

- Write a program module that will save a list of names on a cassette tape. Use "LAST NAME" to indicate the final record. Do NOT use commas to separate the last name from the first since the computer uses commas as markers to separate data entries.

Acceptable answer:

```
10 REM*** This program will transfer information ***
20 REM*** INPUTed from the keyboard to the       ***
30 REM*** computer's memory and then to          ***
40 REM*** cassette tape as a sequential file     ***
50 CLS                          : REM clears screen
60 PRINT "Type in name to be stored."
```

```
 70 PRINT "Type in 'LAST NAME' when finished."
 80 INPUT N$                          : REM keyboard to memory
 90 IF N$ = "LAST NAME" THEN 999 : REM exit
100 PRINT#-1, N$                      : REM memory to tape
120 GOTO 70                           : REM loop could go to 60
999 END
```

5) The statement to transfer information from tape to memory is INPUT#-1.

6) Write a program that will retrieve a file of names and an ID number stored on cassette tape and print each record on the screen

Acceptable answer: similar to above or shorter as below:

```
 10 REM*** This program will transfer information ***
 20 REM***  stored on a cassette tape into the    ***
 30 REM***  computer's memory and then display    ***
 40 REM***  record on screen                      ***
 50 CLS                        :REM clears screen
 60 INPUT#-1, N$, ID           :REM  tape to memory
 70 IF N$ = "LAST NAME" THEN 999  :REM exit
 80 PRINT N$, ID               :REM  memory to screen
 90 GOTO 50
999 END
```

NOTE: When you are transfering data to a file, it is NOT necessary to use the same variable name as was used when it was stored, you just must use the same TYPE of variable (character or numeric). For example lines 60 and 80 could have read

```
60 INPUT#-1,Z$, N
80 PRINT Z$,N
```

Applications:

1. Write a program that is user-friendly that will serve as a directory of names and telephone numbers.

Acceptable answer:

```
 10 REM*** Telephone Directory                    ***
 20 REM*** NAIM$ = name (NAME is reserved word)***
 30 REM*** NMBR$ = phone number with hyphens    ***
 40 CLEAR 2000                      :REM clear more if needed
 50 DIM NAIM$(100), NMBR$(100)      :REM array for up to 100 items
 60 REM
 70 PRINT "This program will allow you to create a file of"
 80 PRINT "    up to 100 names and telephone numbers"
 90 PRINT
100 FOR I = 1 TO 100
110 PRINT "Type in the name"
120 PRINT "Type in 'LAST NAME' when you are finished."
130 INPUT NAIM$(I)
140 IF NAIM$(I) =  "LAST NAME" THEN 200
150 PRINT
160 PRINT "Type in the telephone number"
170 INPUT NMBR$(I)
180 CLS
190 NEXT I
```

```
200 REM*** Instructions to user                    ***
210 CLS  : NMBR$(I) = "          "  : REM dummy # for LAST NAME
220 PRINT "Press PLAY and RECORD on the recorder"
230 PRINT
240 PRINT "Press the SPACE BAR to start the exchange "
250 PRINT " of information from computer to tape"
260 I$ = INKEY$ :  IF I$ = "" THEN 260
270 IF I$ = " " THEN 300 ELSE 260
280 REM
290 REM *** memory to cassette tape                ***
300 FOR J = 1 TO I: REM NAIM$(I) = "LAST NAME"
310 PRINT#-1, NAIM$(J), NMBR$(J)
320 NEXT J
330 CLS
340 PRINT "The file is created. Rewind the tape."
999 END
```

2. Write a program to read in the telephone file created above and search for a given name and then print the telephone number associated with that name.

<u>Summary:</u> What is the difference between a data record and a data file?
What are three advantages to using a file?
What is a sequential file?

<u>Exercises:</u>

1. Write a program that will store INPUTed pupils' names and averages on a cassette.

2. Write a program that will retrieve pupils' names and averages on a cassette.

3. Write a program that will retrieve 20 pupil's names and averages from a cassette tape and then store them in arrays. Write a sort routine that will print out the list in order from the pupil with the highest average to the pupil with the smallest average.

4. Create a file that will contain 25 names and telephone numbers.

5. Write a program that will take an INPUTed name and then read and search the telephone file created in (4) for the name and then print on the screen the name and telephone number.

6. Write a program that will file 50 bromides, maxims or proverbs as records in a file.

Section

2

SPREADSHEET MANAGEMENT

Topic 25
INTRODUCTION TO VISICALC

Performance Objectives: Students will be able to:

- learn the techniques of writing on the VisiCalc spreadsheet
- familiarize themselves with the use of the arrow keys, the space bar, the escape key, and the return or enter key
- differentiate between the procedures used to create labels and values
- understand when to use the GOTO command and when to use the repeat key
- locate the EDIT ENTRY LINE and the PROMPT LINE
- use the VisiCalc program to create, save, retrieve, and delete a text file
- use the VisiCalc program to initialize a data diskette
- become familiar with the STORAGE and the CLEAR COMMANDS
- enter and leave the VisiCalc Program

Vocabulary: New: VisiCalc, spreadsheet
Review: data, text file, initialize

Motivation:

- The SuperData Corporation has offered our class $3.75 per hour to enter data using an electronic spreadsheet for the purpose of creating mailing lists. Furthermore, you will be paid time and a half if you work on Saturday and double time for Sunday.

- What is an electronic spreadsheet?

- What are the names of some that are currently on the market?

- Do large companies really pay salaries to people for entering data?

- If our class were really involved in a project like this, could we use this spreadsheet for any other purpose?

Aim: How do we use the VisiCalc spreadsheet?

Development:

1. Instruct students how to enter the VisiCalc program. Have them become familiar with the term electronic spreadsheet. Point to the cursor and teach the manipulations with the arrows and the space bar. Find the C in the upper right hand corner and describe its function. Practice this technique. Examine the extent of the sheet.

2. Introduce the EDIT ENTRY LINE. Typing on the computer will cause a display on this line. We can make our cursor jump to a new position by using the "›" key. This is similar to a BASIC GOTO statement. ›AB10 will cause the cursor to move to position AB10 on the spreadsheet. Return to A1 by typing "›A1".
3. Begin making entries on the sheet.
   Enter the days of the week on the top line in the second through eighth

column. Show the position of the entry on the EDIT ENTRY LINE. Discuss the manner in which corrections can be made using the escape key for single characters, or the ctrl-C for complete entries. Show how the entry moves from the EDIT ENTRY LINE to the position where the cursor was originally placed. Discuss the use of the return key or the arrow keys for signaling the end of an entry. Complete the days of the week. Enter several names in the first column.

4. Discuss how VisiCalc interprets words beginning with letters as LABELS and those beginning with numbers as VALUES. Explain the unavailability for alphanumeric entries that begin with a number. Demonstrate how VisiCalc sounds a bell to alert us to an illegal entry. Enter numbers in each of the seven rows beneath the days of the week to simulate a weekly report of hours worked.

5. Teach the STORAGE COMMAND. For now we will save this sheet as a text file on a data diskette that has already been initialized. We remove the VisiCalc diskette from the drive and insert a data diskette that has been previously prepared. Above the EDIT ENTRY LINE is a large white line that is called the PROMPT LINE. when we press the "/" key we see a string of capital letters on this PROMPT LINE. Each one of these letters represents a separate COMMAND. Choose the letter "S". Discuss the use of the STORAGE COMMAND and its six options. Select the option "S" and save the file. Use the name "WORK-HOURS".

6. Teach the CLEAR COMMAND. Select /C for CLEAR. Since this command erases all your hard work, VisiCalc gives you the option to think over what you are doing by prompting you to type "Y". Attempt to retrieve our file with the STORAGE COMMAND. You are familiar with this command because you used it to save the file. Type "/S". This time we choose the option "L" to LOAD our file. We see on the PROMPT LINE the question, "File name to load?" Type in "Work-Hours", just as you saved it. Discuss the use of the arrow keys when loading files.

7. Clear the screen with the "/CY" And access the STORAGE COMMAND again. We have already used two of the options. The last two "Q" and "#" we will use in just a moment to leave the VisiCalc program. The "D" is used to delete any unwanted files and the "I" is used to initialize our data diskettes. Now leave the VisiCalc program with "Q". If you have another diskette in the drive you can activate it with "#6", or the number of the slot where your drive is located (if it is not 6).

Applications:

1. Create a list of business applications for VisiCalc. Include inventory, employee records, accounts receivable, etc.

2. Discuss uses of VisiCalc in the home. Include income tax records, cataloging expenses, budgeting, reconciling checking accounts, etc.

3. Create a list of school uses for VisiCalc. Include student, class and department averages, test analysis, record keeping, etc.

4. Discuss how other professions use VisiCalc -- patient and client records, appointment schedules, available medication, blood types on hand, operating room schedules, court appointments, case records, fee

schedules, commission records, individual portfolio accounts, etc.

5. Enter the VisiCalc program and create a text file of students and their official classes. Save the text file for the purpose of sorting in a later lesson.

Summary:

1. How do we enter the VisiCalc program?

2. How do we leave the VisiCalc program without turning the computer off?

3. Can text files be saved on the VisiCalc disk?

4. Is there a quick way to prepare a diskette for saving our files?

5. What do we call the process of preparing this type of a diskette?

6. How do we correct typographical errors?

7. How do we recover from accessing the wrong command?

8. What command do we use to save a text file?

9. Are there any illegal ways to name our files?

10. How is the worksheet cleared? Why is it necessary to clear the worksheet?

11. What can we do if we wish to edit or append a previously developed file?

12. What if we forgot what we named our file?

13. What command causes the cursor to jump to a new location? How do we access that command?

14. What happens if we type over an existing entry? Is this procedure ever useful?

Assignments:

1. Initialize a blank diskette to be used as your VisiCalc data disk.

2. Create and Save a file of all your grades in high school. Include at least 5 subjects and at least 4 grades in each subject. Do not use the same number of grades in every subject.

3. Describe how VisiCalc interprets the following characters:
   a) "3" b) "2Y" c) "A" d) "/" e) ">"

Assignment for More Advanced Students:

1. VisiCalc files are compatible with several word processors. Try saving a file and then loading it into one of your word processors. You can manipulate it in any manner that you wish, and then print it.

Topic 26
USING VISICALC FORMULAS AND COMMANDS

Performance Objectives: Students will be able to:

- use the VisiCalc formulas such as SUM and AVERAGE as well as student supplied formulas
- appreciate the speed at which VisiCalc calculates using the REPLICATE command
- make hardcopies of their worksheets using the PRINT command
- understand the meaning and use of right and left justification as they apply to numbers and words by using the FORMAT command
- examine the GLOBAL FORMAT command and see the effect of making changes on the entire worksheet as opposed to only selected columns
- make selected areas of the worksheet stationary to facilitate ease of entries

Vocabulary: New: replicate, justify, default, scroll, global
Review: format, hardcopy, prompt

Motivation:

- What character is used to access the VisiCalc commands?
- What do these letters represent?
- How do we "teach" VisiCalc a formula?

Aim: How do we use VisiCalc to do calculations?

Development:

1. Review the procedure of entering the VisiCalc program. Discuss making back-up copies of data disks. Have students explain procedure for loading the file created on the previous lesson. Teacher should prepare several files to facilitate class discussion, save time and demonstrate different techniques of loading. (I.e. Using the arrow keys to search for a file as opposed to printing file name.)

2. Work-Hours file should have been expanded to contain 40 names. The effectiveness of the replicate command is best demonstrated with long lists of data.

3. Establish need to develop procedure to calculate wages earned for the forty employees based on a weekday salary of $3.75 per hour, time and a half on Saturday and double time on Sunday.

4. After a brief discussion of the algorithm involved introduce the @SUM FORMULA. Discuss the selection of the next sequential column and the creation of appropriate labels. Discuss the limitation of the nine character column but do not change the size at this time. Position the cursor on the next vertical position, (I2). Type @SUM. Where are the figures located that we wish to total? Enter these coordinates on the Edit Entry Line. Your entry should look like this: @SUM(B2...F2). Now press return and see the answer appear in coordinate I2. Discuss

SUMming non-consecutive values using the comma. (For example, if we wanted to find the Sum of the hours worked on Monday, Wednesday, and Friday we would type @SUM(B1,D1,F1)). How would you find the SUM of Tuesday, Friday, and Sunday? What am I prompting for with @SUM(B1...D1,F1,X1)? Introduce other VisiCalc commands that work in the same manner as @SUM: @AVERAGE, @COUNT, @MIN, and @MAX.

5. Introduce the REPLICATE COMMAND. Discuss the meaning of the word replicate. Have students predict the results of using this command. Call students attention to the display that occurs after entering /R. Carefully describe the difference between SOURCE RANGE and RETURN. What procedure do we wish to repeat? Stress the use of the period key to separate the desired range. Show the formula as it appears on the prompt line. Read the message (REPLICATE:N = NO CHANGE, R = RELATIVE). Since each employee worked a different set of hours, it is that SUM that we are looking for. Press the R key as many times as we have coordinates in our formula. In this case two times, B2 and F2. What happens after the second R is pressed? The teacher should move the cursor up and down the column pointing to the formulas that appear on the PROMPT LINE. We will return to the REPLICATE COMMAND when we calculate the wages.

6. Now we are ready to make up our own formula. Label J1 wages and position the cursor on J2. Keys such as +,-,(, and * are considered values by VisiCalc. After developing the formula type in:

   (H1*3.75)+(F1*3.75*1.5)+(G1*3.75*2).

   Use the replicate key to complete all wages and review procedure for its use. Move the cursor through several entries so that students can see the relative changes in the formula for each employee. An examination of the wages column should lead students to realize the need for displaying a dollar-and-cents format.

7. If difficulty arises as to the correct rows and columns being accessed, the WINDOW command and the TITLE command could be introduced at this time. In slower groups these two commands could be eliminated without disturbing the continuity . /W followed by the options H V 1 S U is known as the WINDOW or split screen command. The options are H (horizontal), V (vertical), 1 (return to one screen, no window), S (scroll both parts of the screen in a synchronized manner), U (unsynchronized, scroll only one part of the screen). The screen is split either horizontally or vertically at the current cursor position. The semicolon causes the cursor to jump over the window.

8. The TITLE COMMAND accessed by /T has the options H (horizontal), V (vertical), B (both), N (negate). This command freezes all rows at and above the current cursor position when the H option is chosen and, at or to the left of the current cursor position when the V option is chosen. This is easily demonstrated by using the arrow and the repeat keys.

9. Introduce the FORMAT COMMAND. /F has the following options: D (default to restore original display), G (general labels left, numbers right), I(integer), L (left justification), R (right justification), $ (dollars and cents), # (for DIF files). Position the cursor over I2 and type /F$, when return is pressed 208.125 will change to 208.13. Review the

REPLICATE command and change the format of the entire wage column.

10. Introduce the GLOBAL COMMAND. Show students how the use of REPLICATE can be avoided by just using "/GF$". Point out that everything has now taken on a dollar and cents format. Have students explain the disadvantage of this procedure. Discuss situation where the GLOBAL COMMAND could be very advantageous. If time permits, the use of "/GC" to change column widths can be introduced.

11. Introduce the term hardcopy and the command P for PRINT. Type /P, VisiCalc prompts with PRINT: FILE, PRINTER, #(OF SLOT). Respond with 1 as that is the slot where our printer in located. VisiCalc prompts RETURN TO CONTINUE, press return. VisiCalc prompts, LOWER RIGHT SETUP. Since we want the entire worksheet select J41. As soon as we press return the printer is activated and our file is printed. At this point the method of positioning the cursor over the lower right coordinate instead of typing in the name of that coordinate should be discussed.

Applications:

1. Review the /SS command by saving the worksheet just created.

2. Review the /SL command by loading the previously prepared list of students and grades.

3. Use the @AVERAGE function to calculate student averages.

4. Use the @MIN function to determine the lowest grade.

5. Write a formula that will calculate student averages by eliminating the minimum mark.

6. Write a formula that determines a weighted grade counting 75% class average and 25% Regents grade.

7. Load the Work-Hours file and calculate the SUMs vertically, this time finding the number of hours worked each day.

8. Assume the firm SuperData agrees to pay the class an additional $1.00 for each $4.50 earned by the class. Write a formula that can be used to find this number and save the value in J42. Use the "/FI" to round this number to the nearest cent.

Summary:

1. What character accesses the SUM function? What other functions can be accessed by this key?

2. What is the function of the REPLICATE COMMAND?

3. How do we instruct VisiCalc that we are about to write a formula?

4. When working with large worksheets what commands freeze sections in selected parts of the screen?

5. What character is used to jump over the divider when using the the window

command?

6. What is the generally selected format for writing numbers? words?

7. What two commands enable us to select right or left justification?

8. Describe when you would use the FORMAT COMMAND and when you would use the GLOBAL COMMAND.

9. Why is it necessary to make a hardcopy of our worksheets?

10. Describe the method you would use to make a hardcopy.

Assignments:

1. Load the file you created and calculate the grade average in all your subjects. Describe the way VisiCalc handles missing marks. Use the MIN function to determine your lowest grade. How does the minimum function treat a missing grade?

2. Select a profession and determine how VisiCalc can be used to improve its record keeping.

3. Experiment with the "/-" COMMAND. This is called the repeating character command. Find a way to use this command in conjunction with the REPLICATE command to underline large areas of your worksheet.

Topic 27
VISICALC AND BASIC PROGRAMMING—PART I

Performance Objectives: Students will be able to:

- set up a class computer store complete with inventory lists, price sheets and cost accounting
- perfect their editing techniques using the BLANK and EDIT commands
- relocate sections of data using the MOVE command
- insert columns and rows with the INSERT command
- use the DELETE command to erase unwanted data
- differentiate between the delete option of the STORAGE command and the DELETE command that erases unwanted rows or columns
- use the file option of the PRINT command to create formatted files that can be read by a BASIC program
- understand the difference between files created by the STORAGE command and those created by the PRINT command

Vocabulary: New: inventory, freight, unit price, packaging, gross
Review: replicate, cost, selling price, profit, loss

Motivation:

1. Our class is going to open a computer store. What are some of the supplies that we should carry so that we can best service our students and our school?

2. What would be some of the ways that we could advertise what we are doing?

3. How can the computer be used to assist us with this project?

4. Do you think that VisiCalc can make any of our jobs easier?

5. What are some of the VisiCalc commands that enable us to manipulate the files that we create?

6. Have we encountered any routines as yet that VisiCalc cannot handle?

Aim: Can Visicalc be used to create files that can be manipulated outside of the VisiCalc program?

Development:

ITEM #	ITEM	DESCRIPTION	PACKED	UNIT STOCK	UNIT COST	SHIP COST	UNIT PRICE
A22304	DISKETTES	SINGLE SIDE SD	DOZEN(12)	36	25.00	2.00	36.45
A22404	DISKETTES	DOUBLE SIDE SD	DOZEN(12)	36	35.00	2.80	51.03
A22305	DISKETTES	SINGLE SIDE DD	DOZEN(12)	36	35.00	2.80	51.03
A22405	DISKETTES	DOUBLE SIDE DD	DOZEN(12)	36	45.00	3.60	65.61
A22622	HEAD CLEANING KIT	CLEANER&2DISKS	DOZEN(12)	12	15.00	1.20	21.87
A22850	STORAGE CASE	HOLDS 50 DISKS	FOUR(4)	12	25.00	2.00	36.45
A22810	LIBRARY CASE	HOLDS 10 DISKS	DOZEN(12)	24	5.00	0.40	7.29
A22860	DISKETTE FILES	SIX LIBRARY CS	THREE(3)	6	25.00	2.00	36.45

A22103	DISKETTE BINDERS	THREE RINGS	GROSS(144)	144	1.00	0.08	1.46
A22905	POCKET PAGES	FIVE PAGES	GROSS(144)	144	1.00	0.08	1.46
C12014	RIBBONS	HALF IN.BY14YDS	SIX(6)	12	1.00	0.08	1.46
C12040	RIBBONS	HALF IN.BY40YDS	SIX(6)	12	8.00	0.64	11.66
C12020	RIBBONS	HALF IN.BY20YDS	SIX(6)	12	6.00	0.48	8.75
P00111	PAPER	HALF IN.GR.BAR	CARTON(1)	5	10.00	0.80	14.58
P00112	PAPER	GRAY RULED	CARTON(1)	5	28.00	2.24	40.82
P00132	PAPER	FOR 132 COLUMN	CARTON(1)	5	26.00	2.08	37.91
P00120	PAPER	FOR 120 COLUMN	CARTON(1)	5	30.00	2.40	43.74
B43005	BOOK	COMPUTER PRIMER	DOZEN(12)	24	5.00	0.40	7.29
E65555	POWER STRIPS	FIVE OUTLETS	DOZEN(12)	12	27.00	2.16	39.37
E96005	SURGE PROTECTORS	WITH POWER STP	SIX(6)	12	22.00	1.76	32.08

1. Develop the suggested inventory file. Show students the need to select stock items which suit the needs of the customer. Explain the difference between the way that the manufacturer packages items for shipping and items for sale. Discuss the difference between cost and selling price. Use the REPLICATE command to calculate freight charge at 8% of cost. Review the FORMAT$ command. Develop the formula that will calculate the selling price at 35% above the cost + freight. Have students do the calculations reviewing the above commands. Care should be taken to list items in order of description so that they can be sorted alphabetically at a later time. Selected items should be left off the original inventory list so that they can be added when the INSERT command is taught. Item numbers and printer paper would be good selections for items to be added.

2. Review the STORAGE SAVE command and save the file INVENTORY. Discuss ways of promoting sales. Elicit development of a catalog sheet to distribute to prospective customers. Have students realize that the inventory file can be edited, thus eliminating the necessity of retyping the entire file. Determine items to be eliminated and those to be saved. Discuss formatting of sheet and value of an alphabetical order presentation. Have students realize that VisiCalc does not have a sort instruction. Tell students that they will soon learn a way of creating files that can be manipulated by a BASIC program. Turn the class' attention to the catalog sheet and teach the DELETE command. Eliminate the Stock, Unit Cost, and Shipping Cost columns using "/DC". What keys must be pressed to delete a row? Does VisiCalc have another delete option? Stress that "/SD" is used to delete files.

3. Discuss items left off INVENTORY FILE and review procedure to retrieve file. Teach the INSERT command and use "/IC" to insert the Item # column. Use "/IR" to insert the rows containing the printer paper. Complete description and all other items, recalculating and replicating where appropriate. Teach the BLANK command for eliminating single entries and show students the procedure for correcting mistakes with the EDIT command. Save the INVENTORY FILE. Point out to students the message on the prompt line, FILE EXISTS "Y" TO REPLACE. Elicit the necessity for this check. Save the file but do not clear the sheet. Prepare the catalog sheet by editing the INVENTORY FILE. Discuss need for clearer titles and delete the title row.

4. Review the PRINT command and print the sheet. Have class examine its appearance and discuss possible improvements. Introduce the FILE

option of the PRINT command. Explain to students that there is a way of saving the file exactly the way that it was printed. Stress that these files cannot be retrieved by VisiCalc and safety dictates that we save these files both ways adding the letters PF to every file saved with the "/PF" command. Emphasize that files saved in this way are compatible with BASIC programs and can be sorted, retitled or reformatted. Create two files CATALOG and CATALOG.PF. Clear the sheet.

5. Elicit other ways that sorting would be helpful. Have students understand the usefulness of having a list in order of item #. Introduce the MOVE command and discuss its uses. Have students load the INVENTORY FILE and use "/M" to place the item#'s in the first column. Name the file Item#.PF and save. Practice moving columns to the right and left as well as rows, up and down. Stress difference in procedure used for left move and down move as opposed to right move and up move. Create a sheet that can be used for daily sales. Discuss procedures used for calculating profit and loss, but leave for class exercise.

6. Preview the next lesson for students. Suggest they brush up on sorting techniques so that BASIC programs can be written with dispatch. Leave students with the assurance that the files they created today will be manipulated by a BASIC program at the next class meeting.

Applications:

1. What items would be included on a sheet of expenses?

2. Explain the difference between gross profit and net profit.

3. Write a formula that can be used to calculate gross profit.

4. What procedure would you use to update the inventory file daily?

5. Load the INVENTORY FILE and delete the shipping cost column. Explain what happens to the unit price column as a result of this deletion. Insert two new books.

6. Make a list of all the different ways VisiCalc edits.

7. Describe the way to move a column from A1 to G1.

8. Name two ways that VisiCalc creates files. Name the file-saving technique that you would use to save each of the following files:

   - A distribution of household expenses that is to be updated monthly.

   - The math department library that we wish to sort according to title, author and subject.

   - The names of the students in our junior class that we wish to sort according to official class, math class and alphabetical order.

   - A current list of all our subjects and grades this term that we wish to append after each test.

. A file of names, addresses and telephone numbers of your friends.

Summary:

1. What features of VisiCalc are used to lighten our bookkeeping load?

2. Why is inventory control necessary?

3. How does VisiCalc aid us in inventory control?

4. Can the INSERT command be used to enable us to merge files?

5. Does the BLANK command make the DELETE command unnecessary?

6. What is the difference between the DELETE command and the delete option of the FILE command?

7. What is the result of "/M B3...K3"?

8. When we use the EDIT command where does this editing take place?

9. How do we save files that we wish to manipulate with a BASIC program?

10. Can VisiCalc retrieve files that are created with the "/PF" command?

11. What can we do to insure that we can retrieve all files that we create?

12. How does VisiCalc warn us that we are about to create two files with the same name?

13. What are some of the reasons that we choose to create files that can be read by a BASIC program?

Assignments:

1. Print a master inventory list that you can use to keep track of the store's inventory when it is your turn to work at the cash register. Make your job easier by retrieving and editing our INVENTORY FILE. Make provisions for a blank column where you can enter the items that you sell.

2. Make a tax chart using VisiCalc's formula function and the REPLICATE command. New York City sales tax is 8.25%.

3. Write a sorting program for numerical order.

4. Write a sorting program for alphabetical order.

Assignment for More Advanced Students:

1. The /PF command has many limitations. It cannot be retrieved by VisiCalc and it does not save formulas. The creation of a special file called the DIF file would be an excellent option for a research topic. You can create these files with /S#S and retrieve them with /S#L. Programs that manipulate these files have been published.

Topic 28
VISICALC AND BASIC PROGRAMMING—PART II

Performance Objectives: Students will be able to:

- review concepts learned to create sequential files
- review error messages and onerr techniques
- review MID$, LEFT$, RIGHT$
- write BASIC programs that will read the files created by VisiCalc
- analyze a list of data to determine range, mean and median
- learn search techniques when analyzing a list of data
- develop skill using the VisiCalc program

Vocabulary: New: range, mean, median, EOF
Review: sequential file, file search

Motivation:

- How can we retrieve the files that we created with /PF?
- Do you think it is possible to print only selected sections of the STORE.PF file without recreating a new file using VisiCalc?
- If we let our students open credit accounts, could we use VisiCalc to keep track of their purchases? Could we use this same file to periodically send them statements of their accounts?
- What are some of the ways we could analyze lists of data?
- Is there anything we would be interested in learning about our test grades?

Aim: How do we write a BASIC program to read the files we created with the VisiCalc program?

Development:

1. Elicit different types of sorting techniques and list on chalkboard. Have students realize that files can only be analyzed after they can be read. Review method of writing a program to read a sequential file. Elicit the benefit of including an error trapping technique when reading lists of data. Develop the following program. This one was written in IBM PC and Tandy BASIC. If you are using a different machine, then you must use your own OPEN and other file management statements.

```
10  REM PROGRAM TO READ A FILE
20  REM CREATED WITH /PF
70  I = 1
80  INPUT "HOW MANY NAMES";N
90  DIM Z$(200)
100 INPUT "WHAT VISICALC FILE?";FF$
120 OPEN,"I",1,FF$
130 IF EOF(1) THEN GOTO 200
140 INPUT#1,Z$(I)
170 I = I + 1
180 GOTO 130
```

```
200  REM THIS SECTION PRINTS
210  FOR K = 1 TO N
220  PRINT Z$(K): NEXT K
230  END
```

Have students use this program to read and list the STORE.PF file.

2. Review the MID$, LEFT$, and RIGHT$ BASIC statements. Remind students that when they wrote the program they set the column width at 9. Now if they wish to read only selected columns they know exactly where to look for the required data. Change line 220 to:

```
220 PRINT MID$(Z$(K),9,18); MID$(Z(K),82,9): NEXT K
```

Students will see only the items and their selling prices listed. Experiment with printing only selected areas of this sheet by changing line 220.

3. Discuss the necessity of having a file that sorts in alphabetical order. Review sorting techniques and develop the following program. This one was written in IBM PC and Tandy BASIC. If you are using a different machine, then you must use your own OPEN and other file management statements.

```
10   REM Read and alphabetize
20   REM VisiCalc /PF files
70   I = 1
80   INPUT "HOW MANY NAMES";N
90   DIM Z$(200)
100  INPUT "WHAT VISICALC FILE?";FF$
120  OPEN"I",1,FF$
130  IF EOF(1) THEN GOTO 200
140  INPUT Z$(I)
170  I = I + 1
180  GOTO 130
200  REM BUBBLE SORT
210  FOR L = 1 TO (N - 1)
220     FOR J = (L + 1) TO N
230        IF Z$(L) < = Z$(J) THEN 270
240        T$ = Z$(L)
250        Z$(L) = Z$(J)
260        Z$(J) = T$
270     NEXT J
280  NEXT L
290  INPUT "DO YOU WANT THE PRINTER ON?";Y$
300  IF LEFT$(Y$,1)<> "Y" THEN GOTO 320
310  FOR K = 1 TO N: LPRINT Z$(K):  PRINT : NEXT K
320  FOR K = 1 TO N: PRINT Z$(K):  PRINT : NEXT K
330  END
```

Create a file of student grades. Use the above program to sort according to first name. Show class how students can be informed of their grades by writing individual letters. For the file:

```
DAVID    ABRAMOWIT 99        100
HEKYUNG  AHN       88        92
```

```
YIN-YEE  CHAN       102      105
ELIZABET FUNG       95       96
MIKY     ISHIDA     96       98
JOHN     KUZIN      105      102
HOWER    KWON       99       102
KLIO     LAW        68       97
DEBRA    LEE        105      99
JAYSON   MOY        84       100
PATRICIA MOY        100      90
NELSON   NG         102      99
CHIN-YU  PEI        76       95
SOMMATTIERAMRUP     90       94
MARK     ROGOWSKY   102      95
VYSHI    SUNTHARAL  94       87
DAVID    UM         100      95
```

Add (or replace) the following lines:

```
310  FOR K = 1 TO N
312     PRINT "DEAR " MID$(Z$(K),1,18) ","
314     PRINT:PRINT TAB(5) "HERE IS A LIST OF YOUR GRADES."
316     PRINT MID$(Z$(K),19)
318     PRINT:PRINT TAB(20)"MR. JONES"
319  NEXT K
320  FOR K = 1 TO N
322      LPRINT "DEAR " MID$(Z$(K),1,18) ","
324      LPRINT:LPRINT TAB(5) "HERE IS A LIST OF YOUR GRADES."
326      LPRINT MID$(Z$(K),19)
328      LPRINT:LPRINT TAB(20)"MR. JONES"
329  NEXT K
```

4. Make changes necessary to create a numerical listing.

```
10    REM Numerical listing
20    REM For VisiCalc PF files
70    I = 1
80    INPUT "HOW MANY NAMES";N
90    DIM V(200)
100   INPUT "WHAT VISICALC FILE?";FF$
120   OPEN"I",1,FF$
130   IF EOF(1) THEN GOTO 200
140   INPUT A
160   V(I) = A
170   I = I + 1
180   GOTO 130
200   REM  THIS SECTION SORTS
220   FOR L = 1 TO (N - 1)
230      FOR J = (L + 1) TO N
240          IF V(L) < = V(J) THEN 280
250          T = V(L)
260          V(L) = V(J)
270          V(J) = T
280      NEXT J
290   NEXT L
295   INPUT "DO YOU WANT THE PRINTER ON?";Y$
300   IF  LEFT$ (Y$,1) <> "Y" THEN GOTO 320
```

```
310  FOR K = 1 TO N: LPRINT Z$(K): LPRINT: NEXT K
320  FOR K = 1 TO N: PRINT Z$(K):  PRINT : NEXT K
330  END
```

Create files with long lists of numerical data and sort. Have the data located in several columns to review the technique of selective sorting. Write a program that changes numerical grades to letter grades. Elicit the type of sorting technique necessary for this procedure.

```
10   REM  CONVERT TO LETTER GRADES
20   DIM Z(200)
30   I = 1
40   INPUT "HOW MANY NAMES";N
60   INPUT "WHAT FILE PLEASE?";FF$
70   OPEN"I",1,FF$
90   IF EOF(1) THEN GOTO 180
100   INPUT A
110   Z(I) = A
120   IF Z(I) = 0 THEN C = C + 1:I = I + 1: GOTO 90
130   S = S + Z(I)
140   I = I + 1
150   GOTO 90
180   REM  THIS SECTION SORTS
190   FOR L = 1 TO (N - 1)
200     FOR J = (L + 1) TO N
210       IF Z(L) < = Z(J) THEN 250
220       T = Z(L)
230       Z(L) = Z(J)
240       Z(J) = T
250     NEXT J
260   NEXT L
270   INPUT "DO YOU WANT THE PRINTER ON?";Y$
280   IF  LEFT$ (Y$,1) <> "Y" THEN GOTO 320
290   FOR H = 1 TO N
300     IF Z(H) > 0 AND Z(H) < 65 THEN B = B + 1: LPRINT Z(H),"FAIL"
302     IF Z(H) > 64 AND Z(H) < 70 THEN U = U + 1: LPRINT Z(H),"D"
304     IF Z(H) > 69 AND Z(H) <80 THEN V = V + 1: LPRINT Z(H),"C"
306     IF Z(H) > 79 AND Z(H) <90 THEN W = W + 1: LPRINT Z(H),"B"
308     IF Z(H) > 89 AND Z(H) < 95    THEN X = X + 1: LPRINT Z(H),"A"
310     IF Z(H) > 94 THEN Q = Q + 1: LPRINT Z(H),"A+"
312   NEXT H
370   CLS
390   LPRINT
400   LPRINT B" STUDENTS FAILED"
410   LPRINT : LPRINT : LPRINT U" STUDENTS RECEIVED A GRADE OF D."
420   LPRINT : LPRINT : LPRINT V" STUDENTS RECEIVED A GRADE OF C."
430   LPRINT : LPRINT : LPRINT  W" STUDENTS RECEIVED A GRADE OF B."
440   LPRINT : LPRINT : LPRINT  X" STUDENTS RECEIVED A GRADE OF A."
450   LPRINT : LPRINT : LPRINT  Q" STUDENTS RECEIVED A GRADE OF A+."
320   FOR H = 1 TO N
322     IF Z(H) > 0 AND Z(H) < 65 THEN B = B + 1: PRINT Z(H),"FAIL"
324     IF Z(H) > 64 AND Z(H) < 70 THEN U = U + 1: PRINT Z(H),"D"
326     IF Z(H) > 69 AND Z(H) <80 THEN V = V + 1: PRINT Z(H),"C"
328     IF Z(H) > 79 AND Z(H) <90 THEN W = W + 1: PRINT Z(H),"B"
330     IF Z(H) > 89 AND Z(H) < 95    THEN X = X + 1: PRINT Z(H),"A"
332     IF Z(H) > 94 THEN Q = Q + 1: PRINT Z(H),"A+"
```

```
334  NEXT H
370  CLS
390  PRINT
400  PRINT B" STUDENTS FAILED"
410  PRINT : PRINT : PRINT U" STUDENTS RECEIVED A GRADE OF D."
420  PRINT : PRINT : PRINT V" STUDENTS RECEIVED A GRADE OF C."
430  PRINT : PRINT : PRINT  W" STUDENTS RECEIVED A GRADE OF B."
440  PRINT : PRINT : PRINT  X" STUDENTS RECEIVED A GRADE OF A."
450  PRINT : PRINT : PRINT  Q" STUDENTS RECEIVED A GRADE OF A+."
```

If time permits, a program that analyzes test results can be written. Discuss range, median, mean and offer the following program to do test analysis.

```
10  REM Test analysis
20  REM For VisiCalc PF files
70  DIM Z(200)
80  I = 1
90  INPUT "HOW MANY NAMES";N
100  INPUT "WHAT VISICALC FILE  ;FF$
120  OPEN"I",1,FF$
130  IF EOF(1) THEN GOTO 210
140  INPUT A
160  Z(I) = A
170  S = S + Z(I)
180  I = I + 1
190  GOTO 130
200  FOR L = 1 TO (N - 1)
250    FOR J = (L + 1) TO N
260      IF Z(L) < = Z(J) THEN 300
270      T = Z(L)
280      Z(L) = Z(J)
290      Z(J) = T
300    NEXT J
310  NEXT L
320  CLS
340  FOR H = 1 TO N
350    IF Z(H) = 0 THEN C = C + 1: GOTO 390
360    IF Z(H) > 99 THEN Q = Q + 1
370    IF Z(H) > 0 AND Z(H) < 65 THEN B = B + 1
380    PRINT Z(H),
390  NEXT H
400  PRINT : PRINT
410  PRINT "RANGE  = "Z(1 +C),Z(N)
420  PRINT
430  MM =  INT ((N - C) / 2) + C + 1
440  IF (N - C) / 2 =  INT ((N - C) / 2) THEN  PRINT "MEDIAN ="(Z(MM) +
     Z(MM + 1)) / 2: GOTO 460
450  PRINT "MEDIAN = "Z(MM)
460  PRINT : PRINT : PRINT "MEAN =       "S / (N - C)
470  PRINT : PRINT "THERE WERE "Q"       100'S"
480  PRINT : PRINT "THERE WERE "B"       FAILURES."
500  END
```

Have students analyze lists of test data.

5. Write a short file simulating students credit accounts. Review formula writing and the REPLICATE command. Suggested file:

NAME	BOUGHT	PAID	DUE
FRED	34.50	25.00	9.50
SALLY	50.00	25.00	5.00
TIM	44.00	25.00	19.00
YUNG	67.00	50.00	17.00
JOSE	12.00	10.00	2.00
JIMMY	57.00	57.00	0.00

Use this file to send account statements to students. Add the lines to the Read File.

```
190 FOR K = 1 TO 7
200    PRINT "DEAR " MID$(Z$(K),1,9)
210    PRINT "TOTAL PURCHASES " MID$(Z$(K),10,9)
220    PRINT "AMOUNT PAID " MID$(Z$(K),19,9)
230    PRINT "BALANCE "MID$(Z$(K),28,9)
240 NEXT K
```

Applications:

1. Open a mail order department. Have students write programs that will list specific sections of INVENTORY FILE. Respond to a query requesting the types of ribbons on hand and their cost.

2. Write letters informing all students who achieved a 100% on a test, of their excellent performance.

3. Employ salesmen. Pay them 6% of their sales. Have VisiCalc determine their commission and then send them letters informing them of their earnings.

4. Send scholarship certificates to all students whose final average was 97% or higher.

5. Use the Name and Address file you created to search for all students who live in a given zip code.

Summary:

1. Why is it necessary to close the file?

2. What are some BASIC statements that enable us to read given sections of a worksheet?

3. How is a median determined?

4. What is the error statement that indicates "out of data"?

5. What is the benefit of using the EOF statement?

6. What are some of the different types of sorts that we may use when writing our programs?

7. Why is it necessary to have a file that sorts numerically (as opposed to only alphabetically) when using VisiCalc?

Assignments:

1. Use the library file that you created and develop three different alphabetical listings -- in order of title, author and subject.

2. Create a VisiCalc file that lists a hospital staff. Include the time and location of their assigned operating room. Write a memo informing each one of them of their schedule.

3. Create a VisiCalc file of all your grades. Find your lowest grade using the @MIN function. Write a program that will read this file and print each one of these grades and the subject it is associated with.

4. Create a VisiCalc file listing household expenses. Use the @SUM function to find the total expenses in each category. Write a program that will print only the columns where the total expense was over $500.

Assignments for More Advanced Students:

1. VisiCalc can be used to find missing parts of oblique triangles. Make use of the @COS function and what you already know about the law of cosines to find the angles of a triangle, given the three sides.

2. Assume you are a stockbroker and create several portfolio accounts for different clients. Write letters to your clients informing them of the status of their accounts. Have VisiCalc calculate their profits or losses and use the @NPV function of VisiCalc to project their future investments.

Section

3

TELECOMMUNICATIONS

Topic 29
ON-LINE DATA BASES

Performance Objectives: Students will be able to:

- describe the advantages of on-line data bases for professionals
- explain the concept of menus and sub-menus
- become familiar with the general procedures of bulletin boards and commercial on-line data bases
- define the terms necessary in the understanding of on-line data base systems

Vocabulary: New: on-line data base, log-on, log-off, information age, gigabyte, menu, sub-menu
Review: telecommunications, modem, on-line

Aim: How can on-line data base systems help the professional and home computer user make decisions?

Motivation:

- Many experts have predicted that our society, with the rapid new technological developments, is at the beginning of a new revolutionary period. They predict that history will show how society has moved from an agricultural to an industrial age and is now in a new era called the information age. Telecommunication will play a very important role in this new age. In each age, certain commodities were vital to success. In the agricultural age, the more land one had the greater the productivity. In the industrial age, the amount of capital one had to invest was of vital importance. In the coming information age, it is predicted that knowledge and the ability to retrieve it will be of great importance.

- In an example given in an earlier lesson, the speed of an on-line encyclopedia has many advantages compared to the relatively lengthy time it takes to publish or retrieve information from an encyclopedia in book form. Similarly, describe how on-line data base systems can be of vital importance for the following professionals:

  - doctors
  - lawyers
  - guidance counselors
  - business investors
  - policemen

Development:

1. What is an "on-line data base?" (A source of information which can be accessed by a computer using such communication devices as phone lines. These data bases are maintained by professionals and there is a fee for using these sources.)

2. What types of on-line data bases are there? (All types. They can vary from specific data bases for physicians to special interest groups such as an Apple Users Group. There are literally thousands of these types of

on-line sources of information, and the number is growing daily.)

3. What type of computer equipment is found at the other end of your modem? (It can vary from another personal computer with a modem and a disk drive to a huge main frame computer with billions of bytes (called gigabytes) of stored information.)

4. What is meant by "log-on" and "log-off?" (Log-on is the procedure of properly entering an on-line connection. Usually a correct identification and password is required. Log-off is the procedure necessary to properly leave an on-line connection.)

5. How do most on-line data base systems work? (Upon successful telephone connection and log-on, a system of menus and sub-menus is given to allow the caller to choose the type of action to be taken. In general, the system has a main menu; the selection of an item from this menu may lead to one or more specific menus known as sub-menus.)

6. What do we mean by the term data base? (This is a large collection of information on one subject. Information can be located or updated in a great many ways electronically.)

Applications:

1. Give a typical example of a log-on and log-off procedure?

   Acceptable answer:

   (We type LOGON)
   WELCOME TO THE "XYZ" ON-LINE DATA BASE!
   WHAT IS YOUR USER ID? (we type ID XYZ 12345)
   SORRY THIS IS NOT CORRECT
   WHAT IS YOUR USER ID? (we now type XYZ 12345)
   WHAT IS YOUR PASSWORD? (microcomputer)
   CORRECT! AND WELCOME TO THE WORLD OF TELECOMMUNICATIONS!
   (We type LOGOFF)
   BYE FOR NOW!

2. Can we outline a specific application of an on-line data base?

   . The following sequence is a sample application of using a commercial data base called "The Source":

   After successful telephone connection and logon.

   WELCOME TO THE SOURCE

   1 USING THE SOURCE
   2 TODAY
   3 BUSINESS UPDATE
   4 THE SOURCE MAIN MENU
   5 WHAT'S NEW
   6 COMMAND LEVEL

   (We select "4" and press return.)

THE SOURCE MAIN MENU

1 NEWS AND REFERENCE RESOURCES
2 BUSINESS/FINANCIAL MARKETS
3 CATALOG SHOPPING
4 HOME AND LEISURE
5 EDUCATION AND CAREER
6 MAIL AND COMMUNICATIONS
7 PERSONAL COMPUTING
8 SOURCE*PLUS

(We select "3" and press return.)

CATALOG SHOPPING

1 BOOKS
2 RECORDS, TAPES, VIDEOTAPES
3 CLASSIFIED ADS
4 COMP-U-STORE

(We select "4" and press return.)

COMPUSTORE SHOPPING SERVICE

1 OVERVIEW/INSTRUCTIONS
2 COMPUSTORE BROWSING
3 COMPUSTORE MEMBERSHIP
4 COMPUSTORE ORDERING&BIDDING

(etc.....)

3. How does the user learn how to use the on-line data base more efficiently? (There are manuals available when you sign up for the service. Also, there are usually aids such as a HELP system command which can assist in what options are available and what they mean.)

Summary:

1. What advantages do on-line data bases have?

2. What do we mean by the terms log-on and log-off?

3. What do we mean by menu and submenus?

4. What types of on-line data bases are there?

Assignments:

1. If you were the president of a computer on-line data base company, describe how you would sell the concept of a computer connected to an on-line data base to provide improved service to hospitals.

2. How would a salesperson use an on-line data base to facilitate making more sales?

3. Using the specific example given in this lesson as a guide, design a menu

and sub-menu for your own on-line data base. Give specific examples of what happens during the first three choices.

4. Using the specific example given in this lesson as a guide, write your own log-on and log-off exchange. Your responses will be in lower case, and response given by the on-line data base will be in capitals.

Assignment for More Advanced Students:

1. Develop a program that will simulate an on-line data base. Include some menus and sub-menus along with a logon and logoff procedure. Require a password to prevent unlawful entry into your on-line data base.

Topic 30

BASIC TELECOMMUNICATIONS — PART I

Performance Objectives: Students will be able to:

- understand the use of a modem to communicate
- explain the relationship between the modem, computer and software
- explore some of the capabilities of modem communications
- define the important terminology associated with telecommunications

Vocabulary: New: baud rate, direct-connect, acoustic coupler, modem, Bell-103, electronic mail, telecommunications
Review: I.C. chips, firmware, software

Motivation:

- The concept of telecommunications is a revolutionary new concept involving the capability of the microcomputer with the availability of the telephone. We are at the threshold of a new way of doing things.

- Using your crystal ball, predict how the home computer connected to the telephone will change:

  - banking
  - shopping
  - mailing letters
  - using a library
  - reading a newspaper

Aim: What capabilities does a modem give the microcomputer?

Development:

1. What is a modem? (From the words "MOdulate/DEModulate." It is a device which converts the computer's digital impulses into sound and transmits them over ordinary telephone lines to another computer.)

2. Telecommunications is a term used to express the two powerful abilities that modems provide:

   - Computers can send or receive vast amounts of data (information).

   - Telephone lines reach virtually anywhere in the world.

3. How fast can information be transmitted over the telephone? (The transmission speed is called the baud rate. The average modem for home use is 300 baud which is approximately 300 words per minute. A 1200 baud modem is available for a somewhat greater cost.)

4. What types of modems are available? (In the past only acoustic modems were available, in which the telephone receiver was placed into a modem with two round cups that held the phone. Today, direct-connect modems use standard modular telphone jacks which plug directly into the modem. This is usually referred to as the Bell-103 telephone communication standard.)

5. At the present time, what is the cost of a modem for home computers? ($150- $500)

Applications:

1. What are some of the capabilities that telecommunications presently provide?

   - Virtually any home computer with a modem can access any of the hundreds of on-line database services available. The information that can be tapped is limitless.

   - Communications between computers open new possibilities. For example, a writer can write a novel on a home computer used as a word processor, and then use the same computer to transmit the novel over the telephone thousands of miles away. Telecommunications are not machine dependent. Thus an Apple computer can communicate with a TRS 80 computer.

   - Electronic mail is a fascinating concept which allows one computer to serve as a post office. Any computer can dial another computer to send or receive mail electronically any time of the day or night. Also, this information can be exchanged immediately and accessed anywhere in the world with a telephone.

   - Every day, new applications for computers with modems are being created and developed. These applications range from computer games to banking.

2. How is a modem connected to a computer? A standard (generally used in designing the connection or interface between computers and modems) is the RS-232 connector. Some computers have an RS-232 connector built into the back of the computer while other computers require an interface card.

3. How does the modem do its thing? (Actually the modem's capability is built into the integrated circuits or I.C. chips of the modem device. This is also called the firmware. However, it is the program or software that allows the users to manipulate the modem. There are dozens of communication software programs which range from simple to complex with a variety of applications.)

4. Another application is the ability to leave one computer on in a remote control mode (through software) and dial this computer from an outside location. The calling computer can run the receiving computer by remote control.

5. Can other means be used besides the telephone lines to connect one computer to another? (Yes, new technology is expanding the capability of telecommunications by replacing the telephone lines with other forms such as FM radio waves, microwaves, infra-red waves, satellite communications, laser light and mobile phone communications.)

Summary:

1. What is the function of a modem?

2. Explain the relationship between the modem, computer and software.

3. What is the baud rate?

4. Describe some capabilities that modems give computers.

5. What types of modems are there and how can computers use them to communicate with each other?

Assignments:

1. Describe several advantages of an on-line (computer with modem) computer encyclopedia compared with a home encyclopedia.

2. With the development of portable computers, many salespersons and executives who travel a great deal find many advantages to using these computers with modems. Can you suggest some specific reasons or examples why this is so?

3. Detail how you would organize a computer users' group using telecommunications.

4. Although telecommunications provide many new opportunities and applications, what problems might society find with this new technology?

Assignment for More Advanced Students:

1. If you were president of a computer company, what kind of telecommunication applications would you suggest for your company's programmers to develop?

Topic 31
TELECOMMUNICATIONS — PART II

Performance Objectives: Students will be able to:

- define telecomputing
- describe how telecomputing is accomplished
- identify what is needed in the telecomputing process
- identify some of the various telecomputing services available

Vocabulary: New: telecomputing, videotex
Review: modem, acoustic coupler

Motivation:

- If you had a computer in your home, how might you put it to use?
- What tasks might be easier to accomplish using the computer?
- Would it be possible to interact with another computer?
- How could this be done if both computers were in the same room? (by setting up a network system)
- How could this be accomplished if the other computer was five miles away?

Aim: What opportunities are available within the field of telecomputing?

Development:

1. What is telecomputing? (Telecomputing is the linking of one computer with another computer by means of a telephone line.)

2. What is needed in order to have one computer talk to another computer by means of telecomputing? (a modem, a cable to connect the modem to the computer in some cases, software, and a phone line)

3. What is a modem? (A modem or MOdulator-DEModulator translates the computer language into tones or sounds that can be understood and transmitted by the telephone equipment. When these sounds are received by the other modem, they are then translated so that the receiving computer can understand it.)

4. What types of modems are available?

   - Direct connect modems. (A modem which uses the same modular plug the telephone uses.) The telephone is disconnected and the modem plug is put in its place.

   - Internal modem. (A modem which is connected within the computer itself. This is generally a plug-in card or board that fits within the computer.)

   - Acoustic coupler. (A modem which consists of a base with two cups that hold a telephone handset securely.)

5. Why is software needed to telecompute? (In order to communicate with other computers, software is needed. Software permits the users to send, receive, store or print out data.)

   . How is this software obtained? (The software may be purchased with specific modems, or it may be obtained through user groups or independently.)

   (Note: A software package entitled MODEM (or depending on the version written and the computer it is written for it may be entitled XMODEM, MODEM7, etc.) may be obtained and copied free of charge because it is public-domain software. It can usually be found through local user groups.)

6. Once your computer is ready to telecompute with another computer, how do you log-on? (Review the specific procedures that are necessary to log-on with another computer.)

7. Beside communicating with neighborhood computers, or with the computers of friends, are there any other types of telecomputation available today?

8. What commercial telecomputation services are available today? (CompuServe, Dow Jones News Retrieval, The Source, etc.)

9. What are these commercial telecomputation services used for? (message leaving and retrieval, on-line conferencing, data bases)

   . Review the uses of each service. How are these services used by industry? How are these services used in the home? What types of services do these commercial systems offer? (games and features; convenient services such as computer banking, shopping by computer, etc.; useful information and research data banks)

Applications:

1. Identify some of the commercial telecomputing services available.

   . Determine what types of service are provided by each.

   . Determine the costs of the specific services.

   . Determine what type of equipment is needed to use specific telecomputing services.

Summary:

1. What opportunities exist for use of a computer with a modem hookup?

2. What equipment is needed to telecompute?

3. What commercial telecomputation services exist today?

Assignments:

1. Draw a diagram which shows how computers may communicate with each other over telephone lines.

2. Compare and contrast the types of modems available.

3. Investigate the features desirable in a modem and determine the modems available with these features, including their capabilities, price, connect features, and special features.

4. Obtain commercial literature from some of the telecomputing services, identify what services are provided, how much the services cost, and who would most benefit from the specific services provided.

5. You are setting up a computer system, including a commercial telecomputing system for a Wall Street brokerage firm. What telecomputing system would you use? Why would this system be best? Who would benefit most from this system?

Topic 32
TELECOMMUNICATIONS: MODEMS — PART III

Performance Objectives: Students will be able to:

- define the various modes that modems can perform
- understand the concepts of auto-dial and auto-answer
- develop a BASIC program which will auto-dial another computer
- develop a BASIC program which will auto-answer a calling computer

Vocabulary: New: auto-answer, auto-dial, originate mode, answer mode, terminal mode, carrier signal
Review: firmware, modem, CHR$(), telecommunications

Motivation:

- Write a program that will write and read a text file of 5 names and phone numbers. (Answer will vary according to the type of computer used.)

  An acceptable answer for the Apple II computer is given below:

```
10   REM  PHONE DATA PROGRAM
20   D$ =  CHR$ (4)
30   FOR I = 1 TO 5
40   READ N$(I),P$(I)
50   NEXT
55   REM  TO WRITE FILE
60   PRINT D$;"OPEN PHONE FILE"
70   PRINT D$;"WRITE PHONE FILE"
80   FOR I = 1 TO 5
90     PRINT N$(I)
100    PRINT P$(I)
110  NEXT
120  PRINT D$;"CLOSE PHONE FILE"
125  REM  TO READ FILE
130  PRINT D$;"OPEN PHONE FILE"
140  PRINT D$;"READ PHONE FILE"
150  FOR I = 1 TO 5
160    INPUT N$(I)
170    INPUT P$(I)
180  NEXT
190  PRINT D$;"CLOSE PHONE FILE"
300  DATA   "TOM JONES","221-4568"
301  DATA    "MARY SMITH","421-4599"
302  DATA     "SUSAN KING","321-6642"
303  DATA      "JOE SKOOB","347-5694"
304  DATA       "MIKE  MONITOR","748-9567"
305  DATA        "GAIL  STORM","573-6793"
```

Aim: How can we control a modem to auto-dial or auto-answer other computers?

Development:

1. Are all modems the same? (Yes and No! At the lowest level, all modems perform the function of converting a computer's digital impulses into sound and transmitting them. However many modems have firmware which make them somewhat intelligent by providing various capabilities or modes.)

2. What are some of the modes that modems are capable of?

   . Originate Mode: the computer making a call.

   . Answer Mode: the computer ready to receive a call.

   . Terminal Mode: allows the computer to communicate with another computer in terminal mode. Thus, what one person types the other can see as well.

   . Auto-dial Mode: the ability of the modem to dial another computer without the user having to dial.

   . Auto-answer Mode: the ability of the modem to receive calls without human intervention.

3. Where are all these modes and how are they controlled?

   . The various modes of the modem are built into the circuitry or firmware of the modem. Any of the modes can be controlled by software which the computer sends to the modem. Many of the popular modems supply the end user with a manual describing the various modes, where they are located in memory and how they can be selected.

4. Is it necessary for the end user to know where everything is in a modem?

   . No. Telecommunications software can keep track of where and when to turn electronic switches on or off. However, there is an advantage to having a good understanding of the basic principles involved.

Applications:

(Since we will be dealing with specific examples of interaction between computer, software and modem, the examples listed below refer to an Apple II computer with the Micromodem II from D.C. Hayes Assoc., Inc. However, the concepts have applicability to other computers and modems.)

1. How can we auto-dial another computer?

   . We learn from the manual that ASCII character 17 is needed to dial a number:

```
10 PR#2
20 PRINT CHR$(17);"1(800)555-1212"
30 END
```

   . Line 10 outputs information to the modem which is located in slot 2.

- Line 20 sends the control character needed to activate the auto-dial firmware built into the modem.

  (Note: The firmware will ignore any non-digit characters.)

2. What will you see on the monitor, assuming a modem is in slot #2 and connected to a phone line?

   MICROMODEM II: 1(800)555-1212

   MICROMODEM II: AWAITING CARR.

3. Where did the words above come from, and what do they mean?

   - The words came from the MICROMODEM II firmware, which is pre-programmed into the modem. The phrase "AWAITING CARR" means that the modem is waiting for a carrier signal indicating a phone connection.

4. How can we use the "Phone Data" program to develop an auto-dial procedure?

   - The following program lines can be added to display the names and numbers:

```
200  REM  DISPLAY
210  TEXT : HOME
220  FOR I = 1 TO 5
230    PRINT N$(I),P$(I)
240  NEXT
```

   - Next we can INPUT our choice:

```
250  PRINT "PLEASE SELECT CHOICE NUMBER";
260  INPUT I
270  Q$ = P$(I)
```

   - To dial a number:

```
280  PR#2
290  PRINT CHR$(17);Q$
```

5. How can we develop an auto-answer program?

   - To wait for a phone to ring and automatically answer it, all that a BASIC program must do is select the modem for input and execute an INPUT command:

```
10   IN#2
20   INPUT A$
30   PR#2
40   PRINT "THANK YOU FOR CALLING!"
```

   - The computer will wait at line 20 until a carrier signal is sent by a calling computer. When it receives this signal, the modem turns on its own carrier signals back to the calling computer, and the program continues executing.

Summary:

1. What are some of the various modes that modems have?

2. Describe the interaction between software, hardware and firmware in telecommunications.

3. How can we write a program to auto-answer?

4. How can we write a program to auto-dial?

Assignments:

1. Modify the "Phone Data" Program to accomodate 20 names and numbers.

2. Describe the advantages of having an auto-dial capability.

3. Describe the advantages of having an auto-answer capability.

4. From the applications discussed today, develop your own telecommunications program that will display a menu to call or answer. If "answer" is selected, the program should display, "WAITING FOR PHONE CALL" and await an incoming call. If call is selected it should display 20 names and phone numbers numbered from 1 to 20. When one of the displayed numbers is then selected it will dial that number.

Topic 33
TELECOMMUNICATIONS — PART IV

Performance Objectives: Students will be able to:

- define the important terminology associated with phone line access techniques
- explain the important considerations necessary for successful telecommunications
- understand some of the variables of communications protocol

Vocabulary: New: protocol, asynchronous, synchronous, download, upload, handshaking, full-duplex, half-duplex, parity
Review: software, baud rate, modem, telecommunications

Motivation:

1. As we have said before, the concept of telecommunications is a new and exciting one. Even though this technology is new, problems that have plagued mankind for centuries will still apply. The Biblical story of the Tower of Babel is an example of an age-old problem of communication. The "failure to communicate" applies to computers as well as humans. Write your thoughts to the following:

   - Considering such factors such as differing computers, modems, baud rates and software programs, can you explain why telecommunications could be difficult?

   - Why would successful communication of one computer to another be important in each of the following areas?

     - banking
     - shopping
     - mailing letters
     - using a library
     - reading a newspaper

Aim: What are some of the important considerations of phone line access techniques?

Development:

1. In telecommunications, what do we mean by the term "protocol?"

   - The consideration of the factors necessary for communications between computers.

1. In telecommunications, what do we mean by "handshaking?"

   - The portion of a communications session in which the sender and the receiver establish each other's identity and establish any further protocols needed.

3. Do we need to establish protocol each time we make communicate with

another computer?

- No, since the protocols can be saved and used in future applications. Again, this is a software application. Most terminal programs can save various protocols for recall or modification.

4. What are some of the factors involved in establishing protocol between computers?

   - Synchronous Communication: A method of data transmission where both sending and receiving modems act simultaneously. At each tick of the CPU clock, a sending modem sends out a data bit, and the receiving modem looks for a data bit to come in. This method is used in high-speed data transmissions, and is seldom used with small computers.

   - Asynchronous Communication: A method of data transmission wherein the sender and receiver do not have to be sychronized. Data may be sent at any time. This is the common method of data communications for most microcomputers.

   - Parity: This is a technique used in communications to detect errors in data transmission. Extra bits are added to information forcing the data to always have an odd or even (odd parity or even parity) number of bits. The receiving device checks for the same parity, and, if not found, an error has occurred.

   - Baud Rate: The number of data bits per second transmitted on a communications link. It is vital that both computers work on the same baud rate.

5. What do we mean by the concepts of full and half duplex?

   - A half-duplex line allows two way communications with only one communications line. An example of half-duplex communications is the CB radio which allows one person to communicate at a time.

   - A full-duplex line provides two way communications by using two communications lines, one going in each direction. An example of this is a two way telephone conversation which allows for communications in both directions at the same time. An important feature of a full-duplex modem is "echo-plex." When a character is typed, it is sent to the distant computer and then is "echoed" back to your computer. Thus, what you see is what the receiving computer sees.

6. What do we mean by the terms "uploading and downloading?"

   - This is the transference of a data file or a program from one computer to another. Transferring a file from a host to a remote computer is called downloading. Transferring from a remote computer to a host is called uploading. This concept gets confusing when the two computers are personal computers. In this case the calling computer is the remote computer, and the receiving computer is the host computer.

Applications:

1. How can we change the duplex status of our communications?

- Many terminal programs allow you to choose the duplex status, either half or full, even after communications have been established.

2. Which duplex status should be used?

   - Try full duplex first and if characters are repeating, then switch to half duplex.

3. Once a communication link has been established with another computer, and the correct protocol has been established, are there other considerations which must be met in order to exchange information?

   - Yes. Again, the software takes control of various procedures which must be followed. For example, in some systems a "Logon" procedure must be followed. This procedure may require pressing a "RETURN" or "ENTER" key followed by a sequence of entering an ID number and/or a password. Many new terminal software programs can store this LOGON procedure and automatically execute these commands as if the user typed it at the keyboard.

Summary:

1. Discuss the importance of proper telecommunications between computers.

2. What is meant by the term "protocol?"

3. What do we mean by "handshaking?"

4. What are the factors in establishing correct protocol procedures?

Assignments:

1. Write a short essay on a catastrophe which could occur in the future as a result of incorrect data transmission from one computer to another.

2. Explain the difference between asynchronous and synchronous communications.

3. Discuss the difference between duplex and half-duplex communications. Can you give other examples of each type of communication?

4. Discuss the role of the terminal communications software in helping the user establish the correct protocol in telecommunications.

Assignment for More Advanced Students:

1. Can you theorize a new protocol which can error-check data transmission over the telephone line?

Topic 34
DATABASE INFORMATION SERVICES — PART I

Performance Objectives: Students will be able to:

- understand how to use the online database information service called THE SOURCE
- understand how to use the menus and submenus of THE SOURCE to retrieve information
- list the types of information available on THE SOURCE

Vocabulary: New: menu, submenu, ID number, password, command level
Review: telecommunications, modem, on-line, log-on, log-off, on-line database

Aim: How can on-line database systems help the professional and home computer user make decisions?

Motivation:

- Pretend that you have just started a computer bulletin board company with unlimited resources. Give your company a name and develop an advertisement promoting the kinds of telecommunications services that your company will provide. Decide what your cost to the end user will be for using your company. These advertisements will be read aloud and the class will vote on the best company providing the most services at the least cost.

Development:

1. What exactly is THE SOURCE? (A worldwide network of electronic communication and information services. This is an on-line database with a vast amount of information.)

2. What steps would you take to use the services of a professional on-line database such as THE SOURCE? (Call the toll free number, pay the fee, dial THE SOURCE using your computer, modem and telecommunications software, use your assigned password.)

3. What are some of the services that THE SOURCE provides? (up-to-the-minute news events, electronic mail, airline schedules, electronic conversations, computer games, shopping, trip planning, and stock market report and analysis)

4. What do we mean by a log-on procedure? (The keystrokes needed to get into the electronic database.)

   - The following is an example of THE SOURCE log-on procedure:

     Connection Successful.

```
SOURCENET
301 401R
TERMINAL=

301 49 CONNECTED
Connected to THE SOURCE

XYZ123 (user 42) logged in Saturday, 13 Oct 84 17:51:28.
Welcome, you are connected to THE SOURCE.
Last login Saturday, 13 Oct 84 15:51:36.
(C) COPYRIGHT SOURCE TELECOMPUTING CORPORATION 1984.
-->

WELCOME TO THE SOURCE

  WELCOME TO THE THE SOURCE ONLINE DATA BASE!

  WHAT IS YOUR USER ID? (we type ID XYZ 123)

  SORRY THIS IS NOT CORRECT

  (note: do not use "ID")

  WHAT IS YOUR USER ID? (we now type XYZ 123)

  WHAT IS YOUR PASSWORD? (microcomputer)

  CORRECT! AND WELCOME TO THE WORLD OF TELECOMMUNICATIONS!
```

5. How can we locate information on THE SOURCE? (There are two basic methods for accessing information: (1) using the menus of THE SOURCE and selecting a number or, (2) by using the command level indicated with a --> prompt). While the second is faster and more direct, the first method is recommended for beginners).

6. How can we use the command level to speed up selection of options? (If we type MENU23 from the main menu for example, we jump directly from the second choice of this menu to the third choice of the next menu.) What would "MENU231" mean?

7. What other system commands can we use? (Typing a "P" will bring you back to the previous menu, and "M" will bring you back to the main menu.)

Applications:

1. How does THE SOURCE compare to the on-line database you created earlier?

2. Why is it advantageous to learn to use the command level? (to save time and money since our phone bill increases the longer we stay on-line)

3. Some of the menus from the source have been copied below. Discuss what you might expect to find for these menu selections.

```
WELCOME TO THE SOURCE
1  USING THE SOURCE
```

2 TODAY
3 BUSINESS UPDATE
4 THE SOURCE MAIN MENU
5 WHAT'S NEW
6 COMMAND LEVEL

(we type 5)

```
     ****************
******   WEEKEND    ******
     ****************
      OCTOBER 13 & 14
```

1 A DAY TO REMEMBER, BECAUSE...
2 YOUR HOROSCOPE FOR TODAY
3 TV HIGHLIGHTS
4 CRITIC'S CHOICE: "Call to Glory" Tumbles in Ratings
5 OPINION FORUM: Viewpoint on the Economy
6 FEATURES: Texas Senate Race Seen As Mean and Nasty Politics
7 PERSONALITIES: Private Eye Solves Mystery; People
8 U.S. NEWS WASHINGTON LETTER
9 SPECIAL REPORTS: American Association of Individual Investors Meeting

Remember: For the latest news of the day, type BULLETIN. For all the sports results, type SPORTS.

(We type M to get to the main menu)

THE SOURCE MAIN MENU
1 NEWS AND REFERENCE RESOURCES
2 BUSINESS/FINANCIAL MARKETS
3 CATALOGUE SHOPPING
4 HOME AND LEISURE
5 EDUCATION AND CAREER
6 MAIL AND COMMUNICATIONS
7 PERSONAL COMPUTING

(We type 2)

```
  *  *  *  *  *  *  *  *
     BUSINESS UPDATE
(Updated Monday through Friday)
        WEEKEND
    October 13 & 14
 *  *  *  *  *  *  *  *
```

1 STOCKCHECK
2 MARKET INDICATORS: Stocks Advance Broadly; Extend Gains
3 COMMODITIES INDEX: (Commodity World News Service Now Online)
4 CURRENCY: Dollar Higher on Strong U.S. Economy
5 FINANCIAL NEWS: The Business Week in Review
6 BUSINESS HORIZON: Roundup of This Week's Stories
7 INVESTMENT TIPS: Donoghue: Now a Better Tax-Deferred Plan Than an IRA
8 THE BOARD ROOM: The Trade Bill - Congressional Politics At

Its Best
BIZDEX: A Guide to THE SOURCE Business Services

(We type M to get to the main menu.)

THE SOURCE MAIN MENU
1 NEWS AND REFERENCE RESOURCES
2 BUSINESS/FINANCIAL MARKETS
3 CATALOGUE SHOPPING
4 HOME AND LEISURE
5 EDUCATION AND CAREER
6 MAIL AND COMMUNICATIONS
7 PERSONAL COMPUTING

4. Discuss what topics you might find if you selected "3" from the main menu?

Actual submenu:

CATALOG SHOPPING
1 BOOKS
2 RECORDS, RADIO CLASSICS
3 CLASSIFIED ADS
4 COMP-U-STORE

Summary:

1. What services does an on-line database such as THE SOURCE provide?

2. Describe the log-on procedure for THE SOURCE.

3. What do we mean by menus and submenus?

4. Explain why it is advantageous to use the command level of THE SOURCE?

Assignments:

1. List and explain the five best features you think an on-line database program ought to have.

2. Develop your own log-on procedure with an ID number and password.

3. If you were designing your own on-line database, what type of command level instructions would you develop to save the user time and money.

4. Describe three types of educational games you would include in your education submenu. Describe how they are played and how they are educational.

Assignment for More Advanced Students:

1. Develop a program that will simulate the menus and submenus of THE SOURCE. Include the log-on procedure.

Topic 35
DATABASE INFORMATION SERVICES — PART II

Performance Objectives: Students will be able to:

- understand the nature of CompuServe
- identify the many services provided by CompuServe
- log-on to CompuServe
- use specific services provided by CompuServe
- discuss the impact CompuServe and other commercial telecomputing services may have on our lives

Vocabulary: New: menu-driven program, logging-on, sub-menu
Review: telecomputing

Motivation:

- What is telecomputing? (Telecomputing is a means of linking a home computer to other computers via an ordinary phone line and a modem.)

- What types of telecomputing exist today? (bulletin boards, commercial telecomputing services such as CompuServe, The Source, Dow Jones News/Retrieval, etc.)

- Who uses these commercial telecomputing services?

- How are the commercial telecomputing services used?

Aim: How may we use CompuServe?

Development:

1. Once the telecomputing system of computer and modem is hooked up, how does one log-on to the outside world? Introduce the procedures for logging on, using a free bulletin board as the service.

   (Note: To find free bulletin boards, check with local users groups, computer stores and computer-related publications.)

   - Logging-on to information services generally follows the following steps:

     - A local phone call will connect the modem to a Tymnet or Tenet phone system.

     - The phone system connects you to the telecomputing system.

     - At this point, you give a connection code and a user identification number. Upon acceptance of this information you are then asked for a password.

       (Note: As you type the password, you will notice that the cursor remains stationary on your screen. This is a security against others learning your password without your permission.)

BEST COPY AVAILABLE

- A welcoming message will indicate that you are now logged-on to the system.

- A menu will then be presented, offering you the choice of many activities. At this point, give specific instructions to access the particular activity the class is to use.

2. Now let us investigate the CompuServe system and eventually choose an activity.

   . What services does the CompuServe system provide to the user? (games and features, on-line conferencing, research data banks, etc.)

   . The main menu of CompuServe contains the following choices:

   - Home services
   - Business and financial services
   - Personal computing services
   - User information
   - Subject index

3. What types of services are provided under Home Services?

   Answer:

   Type in a "1" and examine the choice of services. The screen will appear as follows:

```
                Page HOM-1
HOME SERVICES

1 News/Weather/Sports
2 Reference Library
3 Communications
4 Home Shopping/Banking
5 Groups and Clubs
6 Games and Entertainment
7 Education
8 Home Management

Last menu page. Key digit
or M for previous menu.
```

4. Now,let us select from this list. If we choose Communications, we type in "3" and the following screen appears:

```
                Page PCS-30

COMMUNICATIONS

1 Electronic Mail
(user-to-user messages)
2 CB Simulation
3 National Bulletin Board
 (public messages)
```

4 User directory
5 Access
(public file area)
6 Talk to us

Last menu page. Key digit
or M for previous menu.

5. Investigate the other submenus and determine what services are available in each.

6. This approach to selecting a topic is called a menu-driven system. What does that mean?

   . Is this the easiest way to access a specific group of information? Suppose, for example, that when I accessed CompuServe, I wanted to access CB Simulation. Would it be necessary to go through each of the menus presented, selecting "1" Home Services from the main menu, then selecting "3" Communications from the Home Services menu, then selecting "2" CB Simulation from the Communications menu? Is there an easier and faster way to access CB Simulation?

7. CompuServe will allow the user to go directly to the specific page that may be needed by using a "GO" command. By typing "GO" followed by the specific electronic page number, you will immediately access that specific page.

   . Have the class practice accessing specific items after locating them by using the menu-driven principle of accessing the information.

Applications:

1. Investigate the sub-menus for each item in the main menu and create an index of all subjects available.

   . Determine the initial cost of accessing specific services of CompuServe.

Summary:

1. What types of telecomputing services are available when using a commercial service?

2. How does one log-on to a telecomputing service?

3. What specific features are available through the CompuServe system?

Assignments:

1. Investigate the specific services provided by another commercial telecomputing service. Compare and contrast the features with those of CompuServe.

2. Create a bulletin board showing the sub-menus for CompuServe.

3. Select a specific main menu item, then a specific item from its sub-menu. Determine who would benefit using this particular feature. Identify any qualities you might like to change, add to or delete from this particular feature.

4. Write a short story about life in a home which uses all services provided by CompuServe. Trace a typical day in the life of one of the family members using this service.

   (Note: CompuServe is one of the largest general user services presently available in the field of telecomputing. It provides a little bit of something for everyone--from business information, and the Official Airline Guide, to hundreds of databases, arranged within four categories: Personal Computing, Business and Financial, Home Services and Services for Professionals. Thus, you can make airline reservations, charter a yacht, access Value Line or Standard and Poor's, obtain the latest information on stocks, play adventure games like MegaWars and pay your bills at home. The other commercial telecomputing services provide features similiar to CompuServe and may be investigated in a manner similiar to the way CompuServe has been discussed.

Section

4

THE MAJOR PROJECT

Topic 36
COMPUTER ASSISTED INSTRUCTION — WHAT IS IT?

Performance Objectives: Students will be able to:

- understand the goals of CAI
- understand different formats for the lessons
- understand how the computer can be used to its fullest advantage
- describe possible programming methods for CAI

Vocabulary: New: computer assisted instruction (CAI)

Motivation:

- The teacher should demonstrate computer assisted instruction software for different subjects.

Aim: What are the goals of and different techniques used by computer assisted instruction?

Development:

1. Discuss with students the goals of CAI. These can range from reviewing material, to supplementing a text, to independently teaching a unit. One goal is to review material for purposes of taking a test, such as software for preparing for standardized tests, such as the SAT exam. Another goal might be to serve as a supplement to a text or manual, in order to reinforce concepts with examples. The software can function solely on its own to teach concepts and to provide applications and testing.

2. Have students describe the different formats used by the CAI. This description should include the method of presentation of material, student response, and screen layout. For example, material might be presented in a standardized format, with the question always appearing in the same location on the screen. On the other hand, each screen might need to be individually set up. Student response might range from typing one character (for multiple-choice) to doing extensive computation. The computer might respond to incorrect responses by explaining why the responses are incorrect and by providing hints. Students should notice the placement of questions, descriptions, and diagrams on the screen. Students should be aware that material should be properly centered and be visually appealing.

3. Discuss whether the computer is being utilized to its fullest advantage to distinguish the computer from a book. The computer has the capability of providing more entertainment value and feedback than a book. The CAI might use graphics, sound, computer interaction, scoring, and record keeping.

4. Ask students for the different possible methods of testing that can be used by CAI. The computer can respond to incorrect answers with hints and can evaluate student responses to pinpoint weaknesses and set up a study plan. Records of student scores can also be kept.

5. Discuss with students the possible programming techniques that could be used with CAI. These include loops, files (for data for questions, reading passages, records of scores), and arrays. Sound and graphic techniques appropriate for the age level should be considered.

Application:

1. Have students consider the different methods that might be used by CAI for either elementary or junior high or high school students. What differences would there be in terms of presentation of material, student response, computer response, screen layout, testing, and sound and graphics?

Summary:

1. What are goals of CAI?

2. What are possible formats for presentation of material, student and computer response, and screen layout for CAI?

3. What are different possible methods of testing for CAI?

4. Describe and criticize the goals, methods of presentation of material, methods of student response, methods of computer reaction to student response, screen layout, use of sound and graphics, methods of testing and scoring, and record keeping for two commercial CAI software packages.

5. Describe the programming techniques (loops, arrays, files, sound and graphics) that might have been used for these two CAI software packages.

Assignment:

1. Indicate what an ideal CAI software package would contain for:

   - A math lesson for elementary school students
   - An English lesson for junior high school students
   - A biology lesson for high school students

Topic 37
PLANNING A CAI PROJECT

Performance Objectives: Students will be able to:

- ask the teacher appropriate questions
- gather the material necessary for the lessons
- write an overall plan and algorithms for the program
- set up the questions and format for a testing procedure

Vocabulary: New: line editor
Review: algorithm

Motivation:

- Have students choose a subject area and topics in that subject area for a CAI project. Students should choose a teacher to work with for their CAI project.

Aim: How do you set up a CAI project?

Development:

1. After choosing the subject area and topics for the CAI project, the student should meet with a teacher in that subject area. The student should draw up a list of questions for the teacher about the project. These questions should be about the types of material to be covered, the manner of presentation of the material, the types of responses the student should have to make, the type of computer response, the type of screen layout, graphics and sound appropriate for the age level, and the type of testing and scoring methods. The student should make arrangements with the teacher for students in the teacher's classes to try out different parts of the CAI project.

2. Discuss with the students how to obtain source material for the lessons and tests. This material could include teacher lesson plans, textbooks, curriculum guides, and other software.

3. Students should write an overall plan for the program. This should include algorithms and subroutines for each specific function. Subroutines should be written as independently as possible, so that they can be tested and modified independently of each other. The overall plan is dependent upon the goals of the program, method of presentation of material, and types of testing.

4. Discuss with students the actual programming techniques. The use of arrays, files, and loops should be considered. For example, a program segment (using the Apple computer) to input questions from a sequential file might be as follows:

```
100 D$=CHR$(4)
110 PRINT D$"OPEN QUESTIONDATA"
120 PRINT D$"READ QUESTIONDATA"
130 FOR I=1 TO 20
140   INPUT QUESTION(I):INPUT HINT(I):INPUT ANSWER(I)
```

```
150 NEXT I
160 PRINT D$"CLOSE QUESTIONDATA"
```

- The progam segment to print the questions and allow student response might be as follows:

```
200 FOR I=1 TO 20
210   HOME:VTAB3:PRINT QUESTION(I)
220   VTAB10: INPUT RESPONSE(I)
230   IF RESPONSE(I)=ANSWER(I) THEN VTAB12:PRINT"GREAT.  YOU'RE
        RIGHT":GOTO 280
240   VTAB12:PRINT"SORRY. YOU'RE INCORRECT": VTAB13:PRINT
        "HERE'S A HINT:";HINT(I)
250   VTAB15:INPUT RESPONSE(I)
260   IF RESPONSE(I)=ANSWER(I) THEN VTAB16:PRINT"GOOD.
        YOU'RE CORRECT.":GOTO 280
270   IF RESPONSE(I)<>ANSWER(I) THEN VTAB16:PRINT"GOOD TRY. THAT'S NOT
        RIGHT."
280 NEXT I
```

- A possible program segment to set up multiple-choice tests is as follows:

```
500 DATA "  ",...
510 SCORE=0
520 FOR I=1 TO 20
530   READ QUESTION$(I), ACHOICE$(I),BCHOICE$(I), CCHOICE$(I),
        DCHOICE$(I), ANS$(I)
540   HOME:VTAB1:PRINT QUESTION$(I)
550   VTAB5:PRINT ACHOICE$(I): VTAB7:PRINT BCHOICE$(I): VTAB9:PRINT
        CCHOICE$(I): VTAB11:PRINT DCHOICE$(I)
560   GET RESPONSE$(I)
570   IF RESPONSE$(I)=ANS$(I)THEN SCORE=SCORE+5
580 NEXT I
590 VTAB20: PRINT "YOUR SCORE IS ";SCORE
```

5. Students should either use a line editor or word processor to help them with editing their program. Students will then more easily be able to insert, delete, and move segments of the program.

Summary:

1. What are appropriate questions about your CAI project to ask the teacher you are working with?

2. How can you obtain source material for your project?

3. What are the steps in a program to input questions from a sequential file?

4. What are the steps in a program to set up multiple-choice tests?

Assignment:

1. For the CAI project:

- Write a list of questions for the teacher
- Write the lessons and tests
- Write the overall plan and algorithms
- Write the BASIC program, with subroutines, files, and sound and graphics

Topic 38
SCREEN FORMAT

Performance Objectives: Students will be able to:

- set up the screen for the lessons
- set up a mechanism for student responses

Vocabulary: New: user interface, user friendly

Motivation:

- User interface is defined as the interaction of the user with the input and output, that is the entering of data and computer response. User friendly means that the program is easy for students to use. Students should discuss programming techniques to make the CAI project have good user interface and be user friendly.

Aim: How can the CAI project have good user interface and be user friendly?

Development:

1. The material on a screen should always be centered and the entire screen should be used. Students should not, for example, present all the material at the top of the screen. Graph paper may be helpful for planning the position of material. In order to make it easier to keep track of material on the screen, students should always state in the program the row and column number of each line. For example, students should use VTAB and HTAB (on the Apple computer) at the beginning of each line. If students are to be presented with similar material for many screens (for example, many questions, diagrams, or student responses), then the material should be placed on the same position on the screen. That way the student becomes used to and comfortable with the placement of material on the screen. In order to make the screen easier to read, students might separate different types of material (such as questions and responses) with lines or bars.

2. Write program segments to vary the placement of questions, diagrams, and student responses on the screen. For example, questions might be placed above responses or side by side with responses.

3. The type of student response to questions can vary. For example, multiple-choice questions require a one letter response, while completion questions require words or numerical answers to be filled in. Students need to determine the most appropriate response for the age level they are working with.

4. Write program segments to vary the types of student responses.

5. The computer can respond to student response in different ways. The computer should deliver a friendly message whether the student is correct or not. The computer might explain why the student response is incorrect and suggest a method of studying. Another possibility is that the computer provides a hint to the student and a second chance to try the problem. The computer might branch to a section of the program to

help the student. The student might not be allowed to continue with the program unless he or she answers the question correctly.

6. Write program segments to vary the methods of computer response.

Summary:

1. What are possible methods of placement of questions, diagrams, and student responses?

2. What are possible methods of student response?

3. What are possible methods of computer response?

Assignments:

1. Write the program segment for the screen format for the CAI project.

2. Have students who will use the program evaluate the different screen formats.

3. Write the program segment for the mechanisms for student responses for the CAI project.

Topic 39
ERROR TRAPPING

Performance Objectives: Students will be able to:

- prevent the program from bombing because of inappropriate student responses
- wherever possible, prevent students from entering inappropriate responses

Vocabulary: Review: user interface

Motivation:

- Write a program to input a numeric variable. Describe what happens to the program when you type inappropriate responses, such as letters, many lines of data, etc.

Aim: How can you write a program which deals with inappropriate student responses and does not "bomb?"

Development:

1. The BASIC functions ASC and CHR$ may be used with a GET or INPUT or INKEY$ statement to limit student responses. Consider the following program segment to permit only choices A to D to be typed:

```
10 GET A$ (or INKEY$ on some machines)
20 IF ASC(A$)<65 OR ASC(A$)>68 THEN 10: REM ASCII CODE FOR A IS 65, FOR D
   IS 68
```

2. Write program segments to permit (a) only letters to be typed; (b) only numbers to be typed.

3. In the event that the student types too many characters or otherwise types an inappropriate response, the program should branch to a section which will deal with the response. For example, the following program segment written in IBM PC and Tandy BASIC will clear the screen if the length of the response is greater than 10 characters.

```
10 CLS
20 INPUT A$
30 IF LEN(A$)>10 THEN 10
```

The program might branch (through a subroutine) to another screen which will review appropriate responses.

4. Write a program segment to branch to an explanation screen whenever an inappropriate response is typed in response to an INPUT or INKEY$ statement.

5. The INPUT statement, which may "bomb" from inappropriate responses, may be improved by a sequence of steps so that students can not type inappropriate responses. The INKEY$ statement, which only accepts one character, can be modified so that it functions as an INPUT statement. Consider the following subroutine:

```
110 A$=""
120 B$ = INKEY$: IF B$="" THEN 120
130 IF ASC(B$)=13 THEN 190: REM 13 IS THE ASCII CODE FOR RETURN
140 IF ASC(B$)<65 OR ASC(B$)>90 THEN 120: REM THIS LIMITS RESPONSE
150 IF LEN(A$)=10 THEN 120: REM THIS LIMITS THE STRING LENGTH TO 10
160 A$=A$+B$
170 PRINT B$;
180 GOTO 120
190 RETURN
```

6. The program should contain segments to deal with certain keys, such as the control-reset keys on the Apple 2E or BREAK on various other machines. Commands should reboot the program if one of these is pressed.

Summary:

1. What steps in a program can limit student responses as to the types of characters and number of characters?

2. What steps in a program can prevent inappropriate responses to an INPUT?

3. What steps in a program can prevent keys such as control-reset or control-C from interrupting a program?

Assignments:

1. Write program segments for your CAI project to:

   . limit student responses as to types of characters and string length.

   . prevent the program from bombing in the event the user types the wrong keys.

Topic 40
CREATING A MULTIPLE CHOICE TEST

Performance Objectives: Students will be able to:

- develop appropriate tests
- set up files
- print the results with a printer

Motivation:

- What skills are you testing for your CAI project?

- What possible types of tests might you use for your CAI project?

Aim: How do you set up files and tests for your CAI project?

Development:

1. The data for a test can be entered with READ and DATA STATEMENTS or through a file.

   - Have students write program segments to write tests with READ and DATA statements. An example of a program segment to store and retrieve data on a file for the Apple computer for a multiple-choice test is:

```
10 D$=CHR$(4)
20 PRINT D$"OPEN TESTDATA"
30 PRINT D$"WRITE TESTDATA"
40 FOR I=1 TO 100
50   PRINT QUESTION(I):PRINT CHOICEA(I):PRINT CHOICEB(I): PRINT
       CHOICEC(I):PRINT CHOICED(I):PRINT ANSWER(I)
60 NEXT I
70 PRINT D$"CLOSE TESTDATA"
80 PRINT D$"OPEN TESTDATA"
90 PRINT D$"READ TESTDATA"
100 FOR I=1 TO 100
110   INPUT QUESTION(I):INPUT CHOICEA(I):INPUT CHOICEB(I):INPUT
        CHOICEC(I):INPUT CHOICED(I):INPUT ANSWER(I)
120 NEXT I
130 PRINT D$"CLOSE TESTDATA"
```

2. Arrays and FOR/NEXT loops can be used to print the questions, diagrams, and student responses on the screen. Students should write program segments to vary the position of the questions, diagrams, and student responses on the screen.

3. The user should be limited in the type of response that can be made to the questions. For example, in a multiple-choice test, the user might only be able to type letters A through D. In a completion test, the user might only type strings of less than ten characters. Students should write program segments to allow different types of responses. The program segment should not allow inappropriate responses such as typing numbers for letters and should not bomb when keys such as control-C or control-reset are typed.

4. The scoring should analyze the user's performance in different skill categories. For example, consider the following program segment to analyze skills according to four categories:

```
10 R1=0:R2=0:R3=0:R4=0
20 FOR I=1 TO 100
30   IF I<10 AND ANSWER(I)=RESPONSE(I) THEN R1=R1+1
40   IF I>10 AND I<30 AND ANSWER(I)=RESPONSE(I)THEN R2=R2+1
50   IF I>29 AND I<80 AND ANSWER(I)=RESPONSE(I)THEN R3=R3+1
60   IF I>79 AND ANSWER(I)=RESPONSE(I) THEN R4=4+1
70 NEXT I
80 HOME
90 HTAB13:PRINT "SKILL ANALYSIS"
100 PRINT"SKILL";:HTAB10:PRINT"TOTAL";:HTAB20:PRINT "RIGHT";:HTAB30:PRINT
      "PERCENT"
110 VTAB5:PRINT"1.";:HTAB12:PRINT"9";:HTAB22:PRINT R1;:HTAB32:PRINT
      INT((R1/9)*100)
120 VTAB7:PRINT"2.";:HTAB12:PRINT"19";:HTAB22: PRINT R2;:HTAB32:PRINT
      INT((R2/19)*100)
130 VTAB9:PRINT"3.";:HTAB12:PRINT"50";:HTAB22:PRINT R3;:HTAB32:PRINT
      INT((R3/50)*100)
140 VTAB11:PRINT"4.";:HTAB12:PRINT"21";:HTAB22:PRINT R4;:HTAB32:PRINT
      INT ((R4/21)*100)
```

5. The user should have the option of having the scores printed. The following program segment is a subroutine for the printer option for the Apple computer:

```
10 PRINT "DO YOU WANT YOUR SCORES PRINTED - TYPE Y OR N"
20 GET A$
30 IF A$<>"Y" AND A$<>"N" THEN 20
40 IF A$="N" THEN 100
50 PRINT:PRINT CHR$(4)"PR#1"
60 REM GO TO THE SEGMENT OF THE PROGRAM WHICH PRINTS THE SCORES AND
   ANALYSIS ON THE SCREEN
90 PRINT CHR$(4)"PR#0"
100 RETURN
```

Summary:

1. What steps in a program are needed to write data to and read data from a file?

2. What steps in a program are needed to analyze a score according to skill categories?

3. What steps in a program are needed for a printer option?

Assignments:

1. In your CAI project, write program segments to:

   . Set up files or data statements to input data.

   . Print the questions, diagrams, and student responses.

- Limit the type of student response.
- Score the tests and analyze the results according to skill categories.
- Allow a user to have a printout of the scores.

Topic 41
EVALUATING THE PROJECT

Performance Objectives: Students will be able to:

- write a menu for the CAI project
- evaluate the CAI project as to goals, user interface, interest, tests

Vocabulary: New: menu, evaluation
Review: user interface

Motivation:

- If a group of students could use your CAI project, what questions might you ask them to judge (evaluate) the project?

Aim: How do you evaluate the CAI project?

Development:

1. Before evaluating the project, let us add some final touches. What programming techniques might you use to allow the user to easily go from one section of the program to another (as one does in a book)?

2. One technique is to create a menu, that is a table of contents as one of the introductory screens. An example of a menu for the IBM PC or Tandy computer follows:

```
10 CLS
20 PRINT"MENU"
30 PRINT"TOPIC 1"
40 PRINT"TOPIC 2"
50 PRINT"TOPIC 3"
60 A$ = INKEY$
70 A=VAL(A$):IF A<0 OR A>3 THEN 60
80 ON A GOTO 100,200,300
100 PRINT:RUN "TOPIC1"
200 PRINT:RUN "TOPIC2"
300 PRINT:RUN "TOPIC3"
```

3. In order to allow the user to go from one section of the program to another, one possibility is to allow the user at any point in the program to type a key to go to the menu. From the menu, the user can go to any other section of the program.

4. In addition to the key that goes to the menu, other keys that are helpful are the keys that revert to the previous screen and that advance to the next screen. An example of a program segment is as follows:

```
100 GET A$
110 IF A$=CHR$(13) THEN 200:REM CHR$(13) IS THE CODE FOR THE RETURN KEY.
      200 IS THE FIRST LINE OF THE NEXT SCREEN.
120 IF A$=CHR$(32) THEN 10: REM CHR$(32) IS THE CODE FOR THE SPACEBAR
      KEY. 10 IS THE LINE OF THE PREVIOUS SCREEN
130 IF A$="M" THEN 1000
```

```
140 GOTO 100
1000 PRINT:RUN "MENU"
```

5. Students should develop an evaluation sheet, which should include a checklist and room for comments. The following are questions to be included about the ease of use of the program:

   - Are instructions clear?
   - Can students type inappropriate responses?
   - Are different sections of the program readily accessible?
   - Are screens visually appealing?
   - Are questions and responses loaded quickly enough?

   Questions should also be included about the entertainment value of the program. The following questions might be asked about the value of the material in the lessons:

   - Was the material appropriate for the age level?
   - Was the material clearly written?
   - Was the material presented in a logical order?

   Possible questions about the tests might be:

   - Were the questions and choices clearly stated?
   - Did the questions cover all the important material?
   - Did the format of the test make it easy to use?
   - Was there enough time for the test?

Summary:

1. What are the steps needed to program a menu?

2. What keys might you create to allow a user to branch to different parts of the program?

3. What aspects of the program should be evaluated?

4. What are questions you might ask in an evaluation?

Assignments:

1. For your CAI project:

   - Write a menu.

   - Create keys to allow a user to branch to different parts of the program.

   - Create an evaluation sheet which contains questions about the major areas of evaluation.

Section

5

COMPUTERS IN SOCIETY

Topic 42
MEDICAL DIAGNOSIS

Performance Objectives: Students will be able to:

- explain computer assisted diagnosis
- list two advantages of using computers to help diagnose disease
- list two disadvantages of using computers to help diagnose disease

Vocabulary: New: computer assisted diagnosis

Motivation:

- A three year old with pain in a tooth asks for help. Does the mother take the child to the pediatrician or to the dentist?

Aim: How can computers help physicians diagnose illness?

Development:

1. Refer to the motivation and ask why the choice was the dentist. Elicit that since the pain was in the tooth it made no sense to go to the pediatrician. Indicate that on a very elemental level the parent diagnosed an illness. We shall see how like a parent, the computer can be taught to make diagnoses.

2. Indicate that computers are used in many ways by both doctors and hospitals in treating the sick.

3. List the areas of computer assisted diagnosis, patient records, lab tests, inventory and billing as just some of these ways. Indicate that this lesson will deal with computer assisted diagnosis, the next lesson with lab testing and the following one with computer assisted prosthetics.

3. List the following steps used by a doctor in making a diagnosis:

   - taking the patient's medical history (including a family one)

   - identifying the patient's symptoms

   - matching the symptoms to a list of possible diseases

   - listing the diseases in order of descending probability

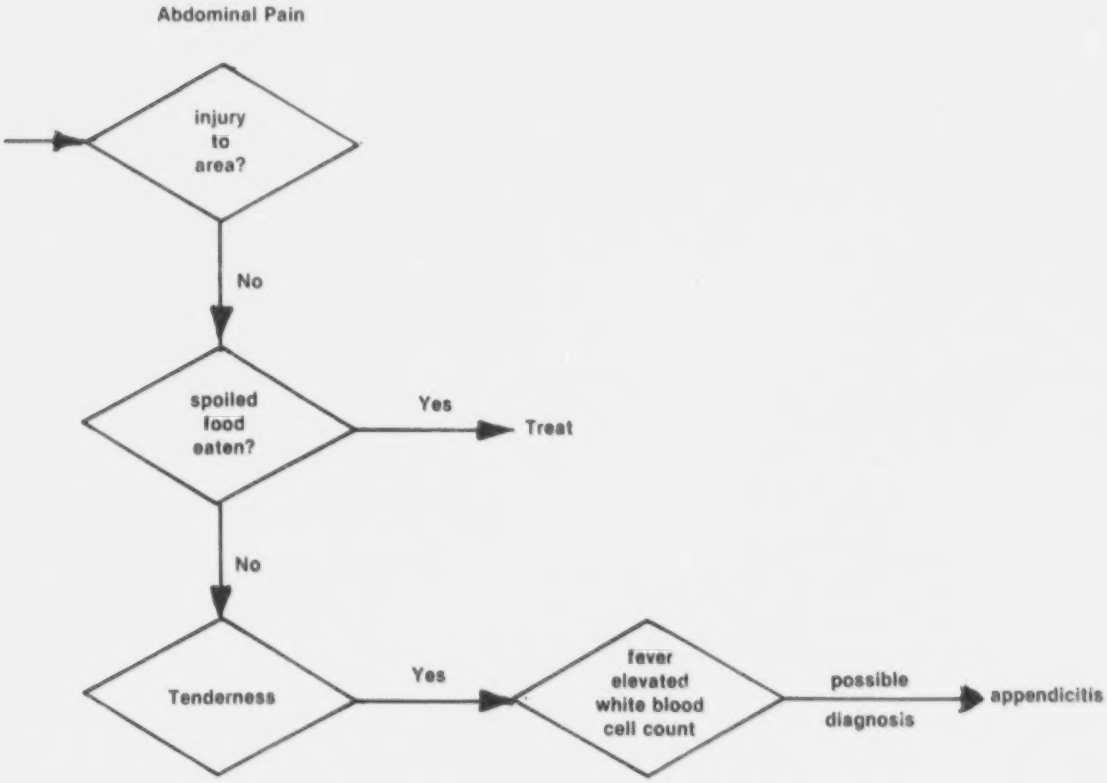
   - choosing the most probable one

4. Describe the case of a person coming to the doctor with pain in the lower right abdominal area.

   - Lead the class in establishing a possible diagnosis of appendicitis by having them ask questions like:

- Was there an injury to the area? (no)
- Did the patient eat spoiled food? (no)
- Is the area hot and tender? (yes)

. Place the diagram below on the chalkboard and use it to show that making a diagnosis involves eliminating possibilities and using those not eliminated as possible diagnoses.



. Ask how the computer could be useful in making the above diagnosis.

. Elicit that the type of reasoning that was just done by the class is exactly the process by which the computer functions and in which it excels.

. Stress that although the reasoning is the same, the line of questioning is not native to the computer since the computer only does what it is told to do.

. Elicit that if the computer were used to diagnose disease, it would have to be programmed with the proper questions to ask and the necessary data to analyze the answers, choose the next branch of questioning, and provide a list of possible diagnoses.

. Elicit that the computer can be as good a diagnostician as the people who programmed it were.

5. Describe the creation of an actual computer diagnostic program.

   . Relate that a programmer was assigned to a master diagnostician. For two years the programmer observed the physician.

   . Each time the doctor examined a patient, took a history, asked a question, ordered a test, or made a diagnosis he was questioned by the programmer.

     - "Why did you ask that?"
     - "What made you order that test and not another?"
     - "What led you to conclude that he had this illness?"

   . The programmer then coded all the doctor's thought patterns and thus a diagnostic program was born.

   . In actual practice the program works well because its logic and its decision-making abilities are those of a master clinician.

6. Describe the operation of a diagnostic computer network which ties together several medical centers located in different areas of the United States.

   . The program prompts the doctor to ask a series of questions which are determined by the responses given to the previous question. As discussed before, the computer prints a list of probable diagnoses and suggests a series of lab tests to be done.

   . The list of probable diseases is arrived at by comparing the symptoms of the patient to a list of past patients' symptoms and their final diagnoses (that which is arrived at after the patient is either cured or dies and an autopsy confirms a diagnosis).

   . In the network, each new case is entered into the database. Included are all the relevant biographical data, the symptoms, tentative diagnosis and lab tests ordered.

   . As lab test results become available they too are entered into the computer. Most importantly, the final diagnosis, as determined by cure or autopsy, is entered and immediately becomes part of the database to be used in diagnosing the very next case.

7. Have the students discuss the advantages and disadvantages of computer assisted diagnosis.

   . Can its use benefit or harm patients?
   . Is the patient being treated by a doctor or by a computer?
   . How accurate is a diagnosis?
   . Is there a possibility of tampering with the database, with individual records or with the course of treatments.
   . Are patients being treated this way serving as guinea pigs?
   . If the computer is doing the diagnosis, ordering tests and prescribing treatment, why do we need doctors?

   . Elicit that in a computer-assisted diagnostic program the computer

circuits take the place of the doctor's brain cells in comparing pieces of data. In most cases, the computer can do this quicker and more thoroughly. Impress upon the students that the logic of the computer is created by the programmer.

8. Discuss the use of computer assisted diagnostic programs in:

   - remote areas not having a doctor
   - submarines on patrol
   - space

9. Discuss other ways in which computers can be used to monitor the quality of health care given to a patient.

   - In some hospitals computers keep track of each doctor's patients including their symptoms, diagnosis and treatment.

   - The computer prints out the information, including a comparison with the most probable diagnosis as stored in its databank and the accepted treatment for this condition.

   - A hospital committee reviews all the information and decides whether the doctor is doing the proper job.

   - Indicate to the students that good diagnosticians, having successfully arrived at a particularly elusive diagnosis, explain, "I had a hunch I can't quite explain, and even though it was not the usual treatment I felt that it was the correct one."

   Ask the students:

   - Can computers have hunches?
   - Does the pressure of computerized peer review stifle a good doctor's fertile imagination and what does this do to the quality of medical care at an institution having such a program?

Summary:

1. How can a computer aid in making medical diagnoses?

2. How accurate is a computer diagnosis?

3. What place does computer assisted diagnosis have in our society?

Assignments:

1. Explain how a computer, which is merely a bunch of electronic ciruits, can diagnose disease.

2. Which is more accurate, a computer's diagnosis or a physician's diagnosis? (Can we ethically entrust a patient's life to a computer?) Explain.

3. How might computer assisted diagnostic programs be helpful to those living in remote areas?

4. How will the availability of computer assisted diagnostic programs affect the practice of medicine?

5. Recently some hackers gained access to a computer in a hospital and tampered with patient records. How much of a danger does this pose to the patient? How can this be prevented? Will the solution to the problem make it too inconvenient for the average doctor to use computer-assisted diagnostic programs?

Assignments for More Advanced Students:

1. Visit a local teaching hospital and find out what, if any, computer assisted diagnostic programs are in use.

2. Describe the programs, how the doctors use them and how they feel about them.

3. Research and describe the diagnostic programs in use at one or all of the following locations:

   . Massachussetts General, Boston, Mass.
   . University of Pittsburgh (INTERNIST program)
   . Stanford Medical Center, Stanford, Calif.

Topic 43
LABORATORY TESTING

Performance Objectives: Students will be able to:

- explain how computers are used to make cross sectional x-rays of the body
- list two ways in which computers are used in medical laboratories to evaluate specimens

Vocabulary: New: acronym, CAT scan

Motivation:

- Display a hardboiled egg. Ask the students to think about how an exact model of the yolk might be made.

Aim: How is the computer used in today's medical laboratory?

Development:

1. Refer to the motivation and ask the students for their suggestions, two of which might be:

   - carefully peeling away the white (albumin) leaving the yolk uncovered

   - cutting the egg into 1-inch thick slices, taking measurements of the the slices and constructing a model

2. Ask them to suggest how this might be done without cutting the egg apart. Elicit that without x-ray vision this would be impossible.

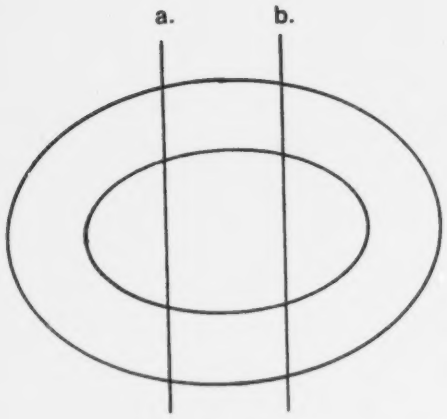
   - State that the problem of determining the dimensions of an egg yolk without cutting apart the egg is similar to the problem faced by doctors when they are asked to determine the extent of internal injuries.

   - Relate that heretofore they have used x-rays which are one dimensional, to show the extent of the damage. This is fine in most bone injuries, but usually unsatisfactory in cases of damage to other types of tissue.

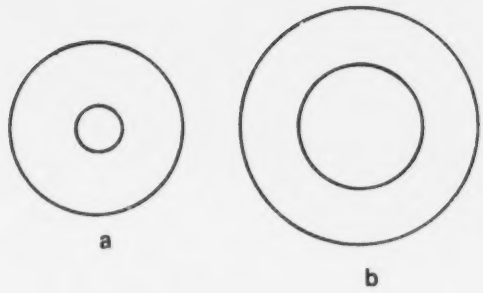
   - Describe the case of the person who develops paralysis of the right arm whose cause is determined to be a tumor located in the brain.

   - In order to determine the feasibility and nature of a surgical procedure, the EXACT shape and location of the tumor must be known. Normal x-rays will show just one aspect; what is needed is a method by which "slices" of the brain might be seen and the exact shape of the tumor with its relationship to other areas determined.

   - To illustrate this, refer to the egg section problem and place the following diagrams on the board.



sliced lengthwise



Cross Sections

- Indicate that the diagram on the left is the egg sliced lengthwise while those on the right are cross sections of the egg taken at two different locations.

- Ask if by looking at diagram A or B one could determine what the whole yolk was like.

- Elicit that one could not because the shape varies over its length. What is needed, therefore, are slices taken every inch which would show the shape as it varied from section to section.

- Explain, therefore, that what the surgeon needs is a means by which an x-ray could be used to take a series of cross-sectional pictures of the skull.

- Show the principle behind x-rays by taking a piece of plastic, taping an opaque object on it, shining a light on the sheet and noting that the shadow of the object is cast on the board located behind the sheet.

- Show that only the shape of the area blocking the light is visible on the screen. The thickness is not demonstrated by the shadow.

- State that x-rays are like the light and are blocked by dense bone or brain tissue, much like the opaque object blocks the light.

- Show therefore, that what was needed was a type of x-ray which could display the area behind the front of the bone (i.e. make a cross section of the skull).

- State that until the advent of computers this was impossible.

3. State that the kind of x-ray machine which makes cross sectional pictures of the body is called a CAT scanner. CAT is the acronym for Computerized Axial Tomography (or slices of the body).

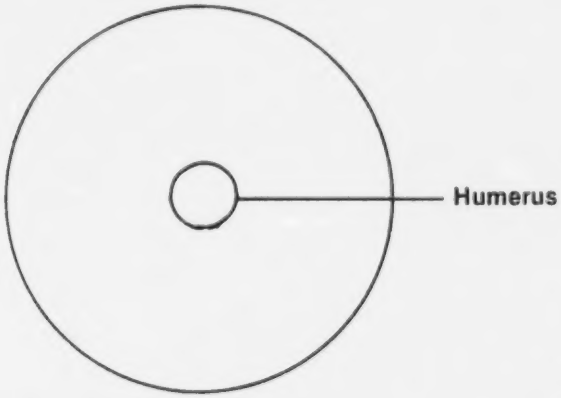
- It provides the surgeon with a set of slices of the skull and its contents which heretofore were obtainable only on autopsy. These slices show the exact size of the tumor, its relationship to other parts of the brain (whether it has invaded nerves and blood vessels or other critical areas).

- Explain that the CAT scanner is the result of the marrying of two technologies, that of the x-ray machine with that of the computer.

- State that a person is inserted into an x-ray machine which is built in the form of a circle. The machine x-rays over a 360 degree area. After an area is x-rayed the patient is advanced a bit and another set of x-rays is taken.

- The key to producing the "slices" is the computer program which integrates all the x-rays taken over one circle and produces the "slice."

- Place the simple cross section of an arm on the chalkboard to illustrate the point.



- Elicit that without the computer there would be no CAT scan. State that while it is most often used to x-ray the skull, it can be used to x-ray all other areas of the body. Its use has allowed surgeons to save untold numbers of lives (particularly of people with head trauma or with other skull involvements).

4. Diagnostic testing. Indicate that besides x-rays other laboratory tests are of value in monitoring a person's health. Among these are chemical tests on blood, urine and other body fluids. Elicit the ways in which a computer might be used to help in these areas. Such ways are:

   - Ordering of the tests. (The requests are entered into a computer which compiles a list for the lab, breaking down the day's work into its individual components. This allows the computer to efficiently plan its schedule of personnel use, machine use and chemical use. It also tells the doctor how much material is needed for each test and its cost. It schedules those tests for which appointments are necessary.)

   - Recording of the results. (The results can be recorded on a printout, sent to the patient's doctor, along with those of all his patients, and

posted on the patient's chart.)

- A bill can be sent to the accounting office for collection. (The computer can also print a bill, have it inserted in an envelope, seal it, stamp it and direct it to the outgoing mail. Naturally, the computer keeps track of those bills outstanding.)

- Performing the test. (The computer could direct the operation of the test equipment, constantly monitoring the machine to see that it is running correctly. This is a huge improvement over manual methods which involve running a known sample at various intervals, screening the results and repeating the abnormal results while the sample is still fresh to assure their accuracy.)

- Describe the actual operation of such a program in which all the hospital's lab instruments are linked to a central computer which automatically monitors each instrument and each test result, calibrates the instruments, prints patient results for the physician, bills for the accounting department and prints the results on the patient's chart. The computer thus has provided quick accurate results, freeing the technician from much of the monitoring of instruments and the lab paperwork.

- Monitoring of patients. (Computers can be set up to constantly monitor patients, signalling the appropriate people in case of problems, thus allowing a more efficient use of personnel and assuring their quick response to an emergency.)

- Computers are excellent tools for comparing large amounts of data and spotting trends. They are thus extremely useful in epidemiological studies.

Summary:

1. What is a CAT scan?

2. How can a computer be used to help in the laboratory?

Assignments:

1. Explain why a computer is necessary for providing a CAT scan. Could a microcomputer be used for this? Explain.

2. What characteristics of a computer make it usable as part of a diagnostic laboratory?

3. What is the danger of a computer making errors in the running of lab tests or the recording of their results? Should legislation be passed to ban their use in this area?

4. What is the danger that unauthorized people will enter a hospital computer system and destroy or change data? Should this possibility be enough to prohibit the use of computers for this purpose?

5. Is having a computer involved in the diagnostic process good or bad for the patient? Explain.

6. After a visit to a local teaching hospital, write a report on how they are using computers in the laboratory.

Assignments for More Advanced Students:

1. Research and report on the details of a CAT scanner.

2. There is a type of CAT scanner which is capable of showing the physician which brain cells are alive and which are dead. Research and report on this.

3. Research and report on the details of an NMR scanner (nuclear magnetic resonator).

Topic 44
BIONICS

Performance Objectives: Students will be able to:

- list three ways in which computers are helping to replace non-functioning organs

Vocabulary: New: quadriplegic, paraplegic, cervical spine, transducer, cornea, pacemaker, DNA

Motivation:

- Tom was in his late teens when he was hit by a car while riding a motorcycle. An injury to his spinal cord made him a quadriplegic, totally dependent on others. Today, thanks to the microcomputer, he is not totally helpless.

Aim: How are microcomputers contributing to modern medical miracles?

Development:

1. Refer to the motivation.

   - Elicit that having damaged the cervical spinal cord (the spinal cord in the neck) the entire body below the neck receives no messages from the brain and can send no messages to the brain. As a result Tom was paralyzed from the neck down.

   - Relate that Tom's brother received donations of computer equipment which he used to build a voice-controlled motorized wheel chair and mechanical arm and voice-controlled home appliances.

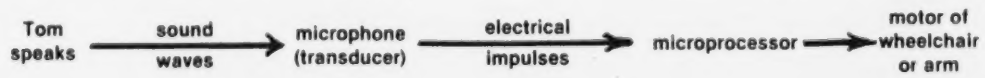
   - When Tom, wished to go forward he called "FORWARD" and the wheelchair moved forward. Similarly, he could have the chair move left, right or backwards at slow or fast speeds. With the attached arm, Tom could bring a glass to his mouth. With the voice-controlled appliances he could watch TV, listen to the radio, type and do some minor chores.

2. Ask, "How can a computer be controlled by a voice?"

   - Elicit that a microphone is a transducer (an instrument which changes sound waves into electricity with each sound creating a different electrical impulse).

   - Elicit that Tom assigned a word or sound to each action he wished to have the wheelchair and/or arm take. The microphone changed these words into electrical impulses which caused the microprocessor controlling the arm or wheelchair to run a program which effected Tom's desires.

3. Elicit the following flow chart:



4. Describe the research aimed at helping a paraplegic to walk.

   . The paraplegic is a person with paralysis in both legs usually due to spinal cord damage in the lower back. Since brain impulses cannot reach the legs the leg muscles cannot function.

   . Walking involves the contraction and relaxation of many muscles and the microprocessor is used to control the strength, duration and order of these external, battery-supplied electrical impulses that trigger these actions. Indicate that the results have been promising with some people taking a few lurching steps, while using rails for support.

5. Describe how blind people are being helped by computers.

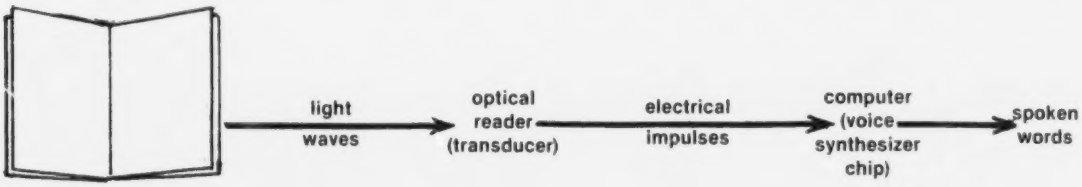
   . Blind people and books:

     - Blind people can access books in two ways: a) reading in Braille (a code of raised dots); b) listening to a recording of a book.

     - In the case of Braille, someone must type everything in Braille, which is an expensive process. Using optical readers, computers can read regular books and print them out in braille.

     - There are many people who can make recordings of novels but not enough people with the ability to make technical book recordings. Again, using the optical reader, a computer can easily make a recording of printed matter.

     - Elicit the following flow chart:



     - Artificial eyes:

       Remind the students of the transducer (something which changes things into electrical impulses). Elicit that all of our sense organs are transducers, changing external impulses into electrical ones understood by the brain.

       Elicit the eye as changing light waves into electrical impulses that are interpreted by the brain. Indicate that most blindness is caused by defects in the eyeball (transducer). Elicit that videocameras are also transducers, doing the same thing as the eyeball does. Suggest that minicameras built with computer-miniaturized technology will, in the future, be connected directly to the nerves or brain thus allowing most blind people to see.

6. Describe a new type of eye surgery for some nearsighted individuals in which the outer layer of the eye, the cornea, is removed, frozen, cut and

shaped by a special machine into an exact pattern allowing it to act like a lens when it is sewn back onto the eye.

- Indicate that the pattern is calculated by a computer and the cutting is controlled by a computer. Without the computer there would be no such operation.

7. Describe the progress made in building artificial joints.

   - Indicate that heretofore, building an artificial hip joint was a long, time-consuming process. However, now the computer is used to analyze and calculate the sizes of the ball and socket of the joint. It even calculates the proper angle that the ball has to have and then controls the lathe which cuts the parts thus allowing extremely rapid and accurate joint construction.

8. Describe the advances made in pacemaker technology.

   - Explain that the heart is a muscle which beats (contracts) when it receives electrical impulses from the brain. At times the nerve connections involved in the transmission become defective and the impulses either never arrive, are weak, or do not arrive at the proper time, thus causing irregular heartbeats and a diminution in the amount of blood pumped.

   - Doctors are able to treat this condition by directly stimulating the heart using an instrument called a pacemaker which is usually placed under the skin.

   - In earlier models, all adjustments regarding strength and timing of the stimulus had to be made before the pacemaker was implanted. Any later adjustments (due to aging of the components or a change in the patient's condition) required an operation and due to the risk, were usually not done. Thus the patient suffered.

   - New pacemakers are microprocessor controlled. In addition to the frequency and timing of the impulses, they also keep a record of how many irregular heartbeats a patient had and/or the number of times the patient needed the pacemaker's assistance. The doctor can externally reprogram the computer and change aspects of the stimulation. Both this and the retrieval of the records can, in some models, be done by telephone.

9. Discuss the role of the microcomputer in replacing some organs.

   - Review the concept that our sense organs merely serve to convert external stimuli into electrical impulses which are sent to the brain for interpretation. Elicit that microcomputer circuitry can serve the same function. Cite the artificial eyes discussed previously.

   - Currently microprocessors are used to help a person control an artificial arm with his brain. The microprocessor is connected both to the nerves which originally served the amputated arm and to motors controlling the arm. The microprocessor changes impulses reaching it from the brain into commands to the motors giving the person control of the arm. In one version, the microprocessor measures changes in

skin electricity and responds accordingly.

- Discuss the artificial pancreas in which a microprocessor constantly monitors blood sugar levels and causes the release of insulin from an artificial sac upon demand. This is akin to the normal person's response. The diabetic thus no longer has to inject himself or match his food intake to the amount of insulin injected or vice versa.

10. Discuss the role the computer is playing in organ transplants.

    - Relate that the big problem in transplanting organs is not surgical. It is the attack of the implanted organ by the host's immune system. This reaction is kept to a minimum by matching the tissue types of donor and recipient (as is done in blood transfusions).

    - Indicate that the time an organ can be kept "alive" after the donor's death is short, and that speed is of the essence. For this reason, a world-wide recipient list is maintained on a computer. Whenever a donor organ becomes available, it is matched to tissue types of those awaiting transplants and dispatched to the closest match.

11. Discuss the use of the computer in genetic research.

    - Relate that the gene is the unit of heredity and it determines our characteristics including the occurrence of certain hereditary diseases such as sickle cell anemia, diabetes and hemophilia.

    - State that the gene is composed of a long chain of four different repeating chemicals. The order in which these chemicals are connected determines the genetic characteristic expressed. Place the following on the board as an example:

      A--T--G--C

    - One gene can have hundreds or thousands of these chemicals. We have been able to synthesize genes by determining the chemical code in a natural gene and then combining chemicals in the determined order.

    - It used to take months to synthesize a gene. Now, thanks to a microprocessor-controlled machine, it is possible to do it within a few hours or, at worst, overnight.

    - State that computers are also used to determine the order of the genes. Today, scientists are implanting human genes into bacteria and having th m produce new substances. (Growth hormone is one of these.) In the future, we will be able to correct genetic defects and computers will help do it.

<u>Summary</u>:

1. How are computers helping paralyzed people?

2. How are computers helping the blind person?

3. How are computers helping to manufacture artificial joints?

4. How are computers involved in controlling pacemakers?

5. How are computers involved in genetics?

Assignments:

1. Explain how a computer is able to respond to a voice command and move a wheelchair.

2. Write a generalized flow chart for having a robotic arm respond to a voice command and give a person a glass of water.

3. Explain why, prior to computers, corneal reshaping operations could not take place.

4. What is it about the computer that enables it to act as a human transducer? List some of the ways it does so.

5. Write a report on the latest advances in computerized medical miracles.

Assignment for More Advanced Students:

1. Research and report in detail on the computer's contribution to modern-day genetics.

Topic 45
COMPUTERS IN THE LEGAL PROFESSION

Performance Objectives: Students will be able to:

- understand how juries are selected
- explain how databases help lawyers select juries
- define circumstantial evidence

Vocabulary: New: circumstantial evidence, peremptory challenges, precedent
Review: database, databanks

Motivation:

- Both you and your friend are to be tried separately for a crime which you did not commit and for which the circumstantial evidence leans heavily against you.

- You have hired an excellent lawyer, and while he will cost you your life's savings, he assures you that by using his new computer service he will probably be able to pick a jury favorable to you.

- Your friend, on the other hand, has no such savings and is represented by a competent lawyer who does not subscribe to the service and who is doubtful as to the positive outcome of the case.

- What does a computer have to do with picking a jury? How would subscribing to an expensive computer service permit your lawyer to feel positive about the outcome of the case while your friend's lawyer, lacking the computer, would feel negative about the same case?

Aim: How are computers changing the way law is practiced?

Development:

1. State that although law has been practiced for many years without the use of computers, the quantum leaps made in computer technology have changed the way the law is practiced.

2. List the areas to be explored: jury selection, legal precedents and office management.

3. Refer to the motivation and ask what the term "trial by jury" means. Elicit that a person being accused of a crime has a right to be judged not by one person but by a whole group of people who are his equals. These people sit and listen to the evidence as presented by the prosecution and by the defense. Then they discuss the testimony, finally arriving at a verdict.

4. Ask why the jury system is better than just having a supposedly impartial judge who is a specialist in the law decide the issue.

5. Elicit that our legal system is designed to be fair to anyone accused of a crime. Elicit that impartiality is rarely one hundred per cent and that the phrase "innocent until proven guilty" would best be served if a person's fate were not left in the hands of one man.

6. Requiring a prosecutor to convince many people of a person's guilt is certainly better for the defendant than requiring a prosecutor merely to convince one person.

7. The group of people a prosecutor has to convince, the jury, is a collection of people - each with their own values, prejudices and faults. Some jurors will be better for a defendant being tried in one type of case than they would be for the same defendant being tried in another type of case.

8. Have the students make the following choices:

   - A person stole food in order to feed his family. Who would make a better juror for the defense - a storekeeper whose store was recently robbed or a person who was once starving himself?

   - A person is being tried for having published secret CIA papers which showed that anti-U.S. rebel forces were overthrowing a government friendly to the U.S. and that the CIA committed illegal acts in order to halt them. Who would make a better juror for the defense - an ex-policeman or a very liberal graduate student?

9. State that in order to provide the best jury for all concerned, the law permits the judge and both attorneys to have a hand in selecting the jury. Each juror is interviewed and any obviously prejudiced individual is not allowed to serve. In addition, each lawyer is allowed a certain number of people he can reject for no reason other than "he doesn't like them." These are called peremptory challenges. The more jurors a lawyer can get who have the same likes and dislikes, the same interests and feelings, the same political and moral convictions, who read the same papers and magazines, watch the same TV programs and go to the same movies as does his client, the more likely the jury is to be sympathetic to his client's plight and the less likely a conviction will be obtained.

10. A lawyer tries to select only those jurors who fit the above pattern. One of the ways this is done is through questioning, a far from perfect method. However, the more background information a lawyer has on each juror, the more accurate his selection will be.

11. Elicit that computers are excellent sources of background information.

12. Explain what a database is.

13. State that massive databanks exist with huge amounts of information on most individuals.

14. Elicit that magazine subscriptions, charge account records, credit ratings, employment history and insurance company records are some of the excellent sources of such information. Elicit how each of the above can affect a person's choice as a juror.

15. Now that computers are cheaper and more powerful, more of the above records can be kept. Elicit that the high-priced lawyer can access an expensive database and obtain information on prospective jurors, thus leading to a better jury selection for his client, while the poorer

person cannot afford to pay for access to such information. Ask the class if the computer has helped or hindered the cause of justice in this case. What can be done to make the situation more equitable?

16. Discuss the areas of jury consultancy and the lawyer's trial tactics. Elicit that many cases are decided by the lawyer's ability rather than by the merits of the case. Indicate that some lawyers are now hiring jury consultants who study the backgrounds of the jurors, note their manner of dress, their speech and their body language. One of their most valuable sources of information is the aforementioned computer databases. After their analysis is complete, they advise the lawyer as to which method of presentation would work best with the jury in the case. Lawyers have testified that the above analysis has actually made the difference between their client being declared innocent or guilty, between a person being sentenced to life imprisonment versus being electrocuted. The high priced lawyer can afford such services. Again, has the use of computers served the cause of justice?

17. Discuss the area of precedent and the law. Explain that law is made in two ways:

    - Laws are written into a constitution or passed by legislative bodies.

    - Existing laws are interpreted to pertain to cases not specifically dealt with in the original law. The resulting interpretation becomes part of the law and is referred to as a "precedent."

18. Every lawyer dealing with a case (either as a prosecutor or a defense attorney) must ascertain not only what the written law says but also must ascertain whether or not there is a precedent in the case. Journals are published listing all court cases and their settlements and research lawyers check all available material for relevancy to a particular case. Even expert, diligent research lawyers overlook things. But with the development of the computer things are changing. All previous law cases are being stored in databases and those able to afford an expensive search service are able to instantly access on their terminals listings of each and every case pertaining to theirs. These listings are accessed by keywords so that the lawyer can be as thorough as needed. These lawyers have a great advantage over the lawyer unable to afford such a service. Again, has the computer best served the interests of justice?

19. Discuss how the computer is changing the law office. Include the following areas: accounts payable, accounts receivable, appointment scheduling, word processors and the customized legal document (in which the word processor can easily customize all legal documents for each individual).

Summary:

1. How has use of the computer affected the practice of law?

Assignments:

1. In what ways has the computer changed the process of jury selection? Has it been affected for the better or for the worse?

2. How has the computer aided law researchers?

3. Has computer-influenced court case outcomes affected the quality of justice for the better or for the worse?

4. How can the computer be used to protect society from parole violators and repeat offenders?

Topic 46
COMPUTER SIMULATION

Performance Objectives: Students will be able to:

- define computer directed simulation
- list five ways in which computers are used in simulating situations

Vocabulary: New: simulation

Motivation:

- A heart surgeon practices on models, animals, and cadavers before operating on humans. How does an airline pilot practice controlling his airplane after an engine failure?

Aim: How can a computer be used to train people by simulating different conditions?

Development:

1. Refer to the motivation and elicit that we can use the computer to simulate airplane flights and airplane part failures; that in this way no plane is lost and no pilot killed while practicing recovery from potentially disastrous situations.

   - Elicit that the computer is used in instructing by simulation in a number of ways:

     - by controlling machines which simulate the real condition, for example, hydraulic airplane simulators.

     - by using high resolution graphics combined with situation programs to depict conditions which a trainee responds to, for example, training automobile mechanics.

     - by simulating events, allowing a trainee to respond to them and then evaluating the response, for example, a medical diagnosis program.

2. Describe an airline computerized pilot trainer.

   - Indicate that it is an airplane cockpit which is mounted on a hydraulically controlled platform with a movie screen for windows.

     - State that a computer program controls the setup. It projects "real" scenes on the screen, controls the sounds in the cockpit and the movement of the platform. The pilot feels that he or she is flying a real airplane.

     - When the pilot is to train for landings, the computer causes the actual scenes the pilot would see on his runway approach to be projected on the window; activates recorders which produce engine noises which sound as they would when a landing is actually taking place; causes the cockpit instruments to mimic the readouts found in actual landings; and causes the hydraulics to move the cockpit in the

same manner that it would move were an airplane making an actual landing.

- The total effect is one of actually landing an airplane.

- Stress that all of the above effects are orchestrated by the computer. The pilot can practice responding to any conceivable condition that might be encountered in the air: engine failure(s), cabin depressurization due to a bullet hole or explosion, loss of a wing. The pilot can practice dealing with each situation until responses become automatic. Without the computer this would be impossible.

- Stress that this setup is so realistic that people can be certified for a pilot's license for certain classes of aircraft merely by putting in enough hours "flying" on simulator and without having ever gone near a real airplane.

3. Describe the use of computer directed simulators in the military.

   . Elicit that they can be used as substitutes for most machines. Have the students list some of the following examples:

   - training pilots to fly

   - training gunners to shoot

   - training tank drivers

   - training navy captains in navigation and docking

4. Describe the Navy's use of an airplane simulator.

   - The Navy has often said that the most difficult and dangerous landing there is for a pilot to make is the night, aircraft carrier landing. The greatest number of naval air crashes occur during this type of landing. By using the simulator, a Navy pilot can practice landings with all the pitching and yawing of a real carrier simulated until he lands perfectly. Then the pilot can try the real thing.

5. Indicate that NASA uses simulators extensively when training its astronauts.

   . Discuss how most phases of a space mission can be anticipated by using computer directed simulators. This includes all possible malfunctions in the system and the proper responses to such malfunctions.

   . Stress that it is one thing to know the text book response to a condition, quite another to react instinctively. Thus, the need for simulators.

   . Cite launches, launch aborts, booster rocket stage separations, orbit adjustments, satellite docking and lunar lander maneuvers as but a few of the conditions which can be simulated.

6. State that programs exist to train people in diagnosing different conditions.

   - Indicate that simulation programs exist which train people in such diverse disciplines as medicine, law, chemistry, and automobile repair, etc.

   - Have the students discuss and identify that characteristic of the computer which enables it to be used to train people by simulation in such a diverse set of occupations as those mentioned above.

   - Elicit that all that must be done is to identify the necessary items to be taught, write a lesson teaching these items, and then write a computer program which presents the material sequentially to the student. Graphics may or may not be used as the lesson dictates.

   - For drill and practice of the above material, simulation programs would be written which would present a case, then ask a question or pose a problem and then allow the student to respond--anticipating every possible student entry. The program would then assess the response and route the student according to the assessment. All of the above, of course, having been previously worked out by the programmer. State that by using high resolution graphics these programs can be rendered highly realistic.

   - Discuss training automobile mechanics. Relate that ten years ago a mechanic could generally do a good job locating and replacing a malfunctioning component. However, today's cars are so complex that the mechanic has to be a diagnostician and not just a grease monkey. The computer serves to train mechanics by simulating car problems and allowing mechanics to diagnose and correct the problems. The computer will assess the response and then simulate either a successful repair or an aggravated condition. The computer can generate every possible condition.

   - Relate how a computer can be used to train supertanker captains. Discuss the fragility of the marine environment and the problems that would be caused by the wreck of a fully loaded supertanker.

   - Relate that navigating a supertanker is an art. Indicate that on some ships when the helm is put over it takes a few miles for the tanker to start to turn. The person piloting the vessel must have many years of experience and practice before being entrusted with such a ship.

   - Discuss the role a simulator might play in training the captain.

   - State that many of our major corporations train their employees by computer-directed simulators.

   - Elicit how simulators could be used in teaching people how to drive an automobile.

   - After hearing the student responses, relate the following:

     - All driving manuals state that to recover from a skid, whether on a

wet road or on ice, one should steer into the skid. State that many of the people you have spoken with know this, yet when the situation arises, they forget the rule and do the wrong thing. Elicit that like the flight simulator, a driving simulator could be used to train drivers to react properly by having them practice skids -- without any danger -- until they instinctively react the right way.

7. Describe how the computer can be used to simulate historical events allowing students to alter their outcomes.

   . Indicate that a program has been written which simulates the events that Lincoln had to deal with as President. It allows students to make decisions that Lincoln had to make and to see the results of their decisions.

   . Describe a program which simulates historical events that happened during the exploration of the West. Again students must make decisions and can "see" the consequences of their decisions. They can then change their minds and again see the difference that their new choice would make.

   . Have the students describe how simulation programs could be written to deal with the following:

     - the projected use of energy in a region with a varying population;

     - the projected increases and decreases in an area's population due to an increase and decrease of manufacturing jobs in the area.

Summary:

1. How is the computer used to train people by directing simulated activities?

Assignments:

1. Find out what happened at the Three Mile Island nuclear generating plant in Pennsylvania. The final accident report indicated that the operators of the plant (those on duty) were not trained well enough and that it was their erroneous response to the conditions that ultimately caused the accident.

   . Explain how computer directed simulator could be used to prevent similar accidents?

2. A computer can be used to simulate automotive problems and thus train mechanics. What is it about a computer which makes it ideal for this type of activity?

3. Describe two areas, not dealt with previously, in which computers are used to train by simulation.

4. How do video arcade games relate to teaching by simulation. Explain.

5. Detail how NASA has used computer-aided simulation as part of their

training program.

6. The movement of airplanes flying through the world's skies is monitored and directed by ground controllers via computers, radar screens, and various microwave transmissions. It takes many years before a controller is ready to work at such busy airports as New York City's Kennedy or Chicago's O'Hare. How can computers be used to help train them? How can computers be used to help maintain order in the skies?

Assignments for More Advanced Students:

1. (Note: A computer having high resolution graphics capabilities is required for this assignment.)

   . Write a program which will simulate an airplane's artificial horizon action during flight:

   - An airplane pilot watches a cockpit guage called an artificial horizon to tell whether the plane is flying level, ascending, descending, or turning left or right.

   - When the airplane turns to the left the "airplane" in the guage turns to the left and it likewise turns right when the plane turns right.

   - The guage "airplane" moves above the line (artificial horizon) when the airplane is climbing and dips below the artificial horizon when the airplane is descending.

   - Write a program which will place an artificial horizon guage on the screen and will have it then respond to the keyboard simulating the actions of a plane as described above.

   - Students having access to joy sticks should also adapt their program to respond to them.

2. Write a report on the Chase Econometrics Report relating it to computer simulation.

3. Write a simulation program which would deal with the increase and decrease of teachers necessary to the student's school as the student population increases and decreases.

Topic 47
DATABASES AND SECURITY ISSUES

Performance Objectives: Students will be able to:

- list three ways in which computers are used to legally invade other people's privacy

Vocabulary: New: database, mailing list

Motivation:

- Helen Radin bought a boomerang through a mail order catalog. A short while later she received letters from the Boomerang Association Inc. and from the Bolo Owners of America inviting her to become a member. In addition, she began to receive boomerang catalogs, ranging from one called "Penultimate Boomerang Accessories" to one entitled "Kangaroo Corner." Various charities began to solicit donations. She was asked to contribute to the "Flea Lovers of America" and even the President of the United States sent her personal letters requesting a donation. Two or three requests and/or catalogs began arriving each day along with phone calls offering to sell her anything she needed and everything she didn't. "Why me and why now?" she wondered.

Aim: How are computer databases being used to invade our privacy?

Development:

1. Elicit that the use of computers eliminated much drudgery. Certain operations, such as record keeping and other monotonous work, are performed faster and better by computers than they could be manually. Point out, however, that while computers are making our lives easier, they are also making it easier for strangers to access both our past and our present. They are making an invasion of our privacy easy.

2. Define the term "database."

   - Use the analogy of a file cabinet drawer which has four folders in it. Folder A contains a list of 100,000 subscribers to 50 different magazines. Folder B has a list of those earning $30,000 or more per year. Folder C contains a list of the population of Brooklyn. Folder D has a list of those people in the U.S. who bought boomerangs last year.

   - Should someone want a list of all the subscribers to a specific magazine they would look in folder A.

   - Should someone want a list of those earning $30,000 or more per year they would look in the second folder.

   - Should someone want a list of those subscribing to "The National Fruitcake Journal" AND earning $30,000 per year they would have to crossmatch the listings in BOTH folders. This would take large amounts of time and be very expensive when dealing with large lists.

   - Indicate that a DATABASE maintains the lists of ALL four folders and

can EASILY print a list of those who fit a certain "profile" made up of ANY combination of the characteristics in the folders. For example, it could print a list of those who subscribe to "The National Fruitcake Journal," earn over $30,000 per year, live in Brooklyn and have NOT bought boomerangs in the last year.

3. Refer to the motivation. Elicit that the reason the catalogs and letters started to arrive was that Helen's name, address, and amount and type of purchase were sold to a mailing list company who entered her name into their database.

   . When someone wanted to sell boomerang accessories they bought the mailing list which would include the names of those people most likely to buy such things (based on their purchase of a boomerang in the past year).

   . Indicate that the mailing list company could have gone a step further and included in their database information obtained from accessing the database of a large credit reporting agency such as TRW which keeps detailed financial information on many Americans.

4. Elicit a definition of junk mail and establish that the names of those who are sent junk mail come from mailing lists like the one described above. Relate that to some people, receiving a box full of junk mail constitutes an invasion of privacy.

5. Have the students describe instances in which their names and addresses were given out by a computer.

6. Relate the existence of an association of mailing list retailers who will remove a person's name from all mailing lists when requested (and who will, conversely, add names to lists).

7. Describe the sophistication of today's mailing lists. Establish that a mailing list works by alphabetizing a list of names and addresses. As the sophistication of the program grows there is a more thorough cross checking of names and addresses eliminating multiple mailings.

   . Have the students recall the times they received three copies of the same letter, each one addressed just a shade differently, but different enough to tell the computer that they were not addressed to the same person.

   . Have the students explain how this can happen. They should include a discussion of the IF/THEN statement and its place in the comparison process and in mailing lists. They should describe how the following things affect the IF/THEN comparison:

     - Use of initials and differently spelled names. For example, even though they were the same person, J. DOE, J.P. DOE and JOHN DOE might be considered three different people by an unsophisticated program and ALL three names would receive mailings.

     - Furthermore, an address of 1875 E. 17 Street and E. 17 Street and Avenue S might be the same address to the mailman but different addresses to the computer.

- The presence and absence of spaces in the strings. For example:

```
LET A$="JOHN"
IF A$="JOHN " THEN PRINT"CORRECT" ELSE PRINT "WRONG"
```

. Explain that when the first mailing list programs were written, computer time and memory were expensive and it was cheaper to multiple mail items than it was to cross check to eliminate duplication. Now however, computer time and memory has become cheaper, postage rates more expensive and highly sophisticated computer mailing list programs exist. A company can direct a mailing to an individual between thirty and forty years of age, whose annual income is between fifty and sixty thousand dollars, who lives in a prescribed three block radius of an exclusive neighborhood, who is interested in boomerangs and who likes to eat cheese.

. Establish that while all this information has been acquired legally it is, in essence, a legal invasion of one's privacy.

8. Refer to the motivation and describe the person who donated money to a charity and who was then deluged by requests. Describe, too, the case of the person who issued a check to a charity and did not put any address on it. The charity got access to the bank's records, got the address and then included the name when they sold their mailing list to other people.

9. Refer to the lesson on the computer and the law and indicate that it is very easy to compile a list of a juror's interests and identify the person as either liberal or conservative.

10. Elicit that magazine subscriptions, charitable contributions, area of residence and type of domicile, credit rating and types of charge purchases, type of car driven and kinds of vacations taken, are but a few of the areas in which computers keeping records can be used to identify an individual.

. Ask if this is an invasion of one's privacy.

11. Show a charge card and relate that many of the larger stores are connected directly to the charge company's computer and that for most purchases, the account number and amount of purchase are entered into the computer for approval. Indicate that on one hand, the ability to instantly check on a credit card protects the consumer from its illegal use in the event it is stolen. On the other hand a penalty is paid, because the computer can later print out, on demand, the time, place and type of purchases made. A person's movements can thus be traced.

12. Stimulate discussion by relating the following cases of invasion of privacy:

. A teacher moved to a new state and applied for a job in a local school. She was turned down when a health insurance company provided unsolicited evidence that she had had an abortion. Invasion of privacy or not?

. A town in the Midwest wished to close a movie theater that showed X-

rated films. The movie theater owner subpoenaed the computerized records of the local cable network which showed that the very same political and religious leaders who were calling for the closing of the theater were paying to watch X-rated films. Invasion of privacy? Protection of a theater owner's civil rights?

13. Library records are increasingly being computerized, greatly streamlining the library's operations. But at the same time the computer is saving money for the library it is making it easier for those few people who would monitor and censor the reading habits of children. Should we stop the use of computers in libraries for this reason?

14. In the following areas, have the class contrast improvements in the quality of their lives made by using the computer against the cost in terms of the invasion of their privacy:

    . Airlines reservations are computerized thus providing instant flight reservation and confirmation even for many months in advance.

        - At the same time it is possible to trace someone's movements by accessing the airline's databank.

    . Telephones run by computers have opened up new worlds in communications with, among others, touch tone dialing, call forwarding and telephone credit cards.

        - At the same time it is possible to look at a computer-generated phone bill listing times and places of calls and trace a person's movements.

    . Personnel files are stored in computers in many big businesses. Has the omnipresence of computer terminals contributed to an invasion of privacy by affording many people access to these records?

    . Extensive computer-generated credit records are kept for most people without their knowledge. Is it an invasion of their privacy to disclose this information to a merchant who subscribes to that service?

    . Does the increased number of records kept which are made possible by the use of the computer have any bearing on the issue?

15. Elicit that the very same capabilities of the computer which make it such a potent tool for efficiently managing the ever growing and essential mass of records, also make it a great tool with which to invade a person's privacy.

    . Do we have to take the bad along with the good? Is it too great a penalty to pay?

16. Describe the capabilities of today's computers in monitoring telephone conversations and computer transmissions which are sent cross country by microwaves. Indicate that by using sophisticated and powerful computer programs "people" can record every conversation they choose to and have

the computer search for key words in the conversations, printing out only those conversations containing these words. Alternatively, they can print out only those conversations coming from certain extensions.

Summary:

1. What is a database?

2. How have those qualities which have made the computer such a time-saver also made it an excellent tool for invading privacy?

Assignments:

1. A divorced parent leaves town and stops sending child support payments. Computer charge card records are used to find the individual. Is this an invasion of privacy or not? Explain.

2. Why do we accuse the computer of invading our privacy when the same records were kept before its advent?

3. Explain how the use of the computer to run a charge card company's operations might lead to an invasion of privacy.

4. What is meant by the term "credit rating." How is it determined? How necessary is it for living in today's society? How easy is it for errors to become part of it? Is it possible for a merchant to ruin a person's credit rating for spite? Do married women have credit ratings?

5. Students should obtain parental permission, find out the name of the local credit reference agency and obtain a copy of their credit record.

6. Use a "coded" name (such as initials) when entering a magazine subscription or when buying something from a mail order firm. See how many pieces of junk mail you get addressed to this name.

7. It is said that computer scientists put great efforts into developing ever denser chips. This makes computers more powerful, drops their price and increases their role in society. At the same time access protection is neglected. Research and report on some of the common methods of protecting the integrity of computer data and the efficacy of these methods.

Topic 48
DEPERSONALIZATION IN THE ELECTRONIC AGE

Performance Objectives: Students will be able to:

- identify possible dangers to individuality from job-related computer use
- identify possible dangers to society in general from computer use

Motivation:

- Display a Hollerith card (a punch card or IBM card). Indicate that these frequently bear the legend, "Do not fold, spindle or mutilate." There was a popular bumper sticker that said, "I am a human being. Do not fold, spindle or mutilate."

- What fear does this joke play upon? (That there is a danger of humans being treated as machines.)

- We can define machines as interchangeable, non-individual units.

Aim: As machines become more like humans, is there a danger that we will be treated more like machines?

Development:

1. IBM uses the Charlie Chaplin character as the advertising symbol for their personal computers, calling them "machines for modern times." But, ironically, in the movie Modern Times, Chaplin was satirizing the dangers to individuality of modern technology.

2. In what ways may computer use make people more like machines on their jobs?

   - Routinization of jobs - many activities are becoming standardized due to requirements of computers. Individual differences become less important than the ability to fit into a large system.

   - Monitoring by computer - computers can act as full-time overseers of workers. In a word processing pool, for example, a supervisor can determine the precise rate at which a typist is working at any moment. A computer records the exact time a telephone operator spends on each call.

3. What effect does this have on the worker?

   - Technological unemployment - many skills categories may be removed from the job market as computers and robots replace human workers, especially in factories. Many semi-skilled workers will have only menial jobs available to them.

4. How will this affect how workers think about themselves? How can this effect be minimized?

5. New technologies frequently require changes in everyday routines and behavior, for good or ill (for example, the factory system required

the ordinary worker to be aware of timekeeping and punctuality). How are people being forced to adapt to computer routines?

- Less human contact in dealing with large institutions. For example, automatic teller machines instead of tellers in banks.

- School programs are determined by computers.

- Telephone machines, both automatic answering machines and automatic message output (such as attendance calls made by schools) are in greater use.

- Can you think of others?

6. The use of numbers to identify a person is becoming more prevalent.

   - How many numbers identify you? (Social security, bank account, student number, credit card account numbers, employee identification numbers, etc.)

7. The difficulty people encounter in ordinary transactions range from errors in forms read by computers, to requiring conformity in details such as the use of # 2 pencils when filling out forms. Can you think of others?

Applications:

1. Do any of these practices actually harm us? If so, which ones and in what ways?

2. If these practices are actually harmful, how can they be justified?

3. What can be done to minimize these negative consequences?

4. Will the growing use and sophistication of computers make these problems greater?

5. Can more powerful computers eliminate some of the problems we discussed? Which ones? How? Which problems do you think will become worse?

Summary:

1. Can you think of a fictional character better suited to represent IBM's modern man than Chaplin's Little Tramp?

Assignment:

1. Write an essay on the subject of computer-generated loss of individuality. Be sure to identify both personal and social implications.

Section

6

OTHER NUMBER SYSTEMS

Topic 49
BINARY/DECIMAL CONVERSIONS

Performance Objectives: Students will be able to:

- determine the place value and digits in base 10 (decimal)
- determine the place value and digits in base 2 (binary)
- determine place value of any base b in terms of powers of b
- determine how many digits in base b
- write binary numbers in decimal form
- write decimal numbers in binary form
- list two reasons why binary system is used instead of decimal system in a computer

Vocabulary: New: binary, digit, bit, place value, base, exponents
Review: decimal

Motivation:

- What is the place value of each digit in the number 345?

  - Elicit three hundreds, four tens, and five ones.

- Review place value of base 10 (decimal system).

Figure 1

Binary

	ten-thousands		thousands		hundreds		tens		ones	
10,000	10	*	10	*	10	*	10	*	1	$= 10^4$
1,000			10	*	10	*	10	*	1	$= 10^3$
100					10	*	10	*	1	$= 10^2$
10							10	*	1	$= 10^1$
1									1	$= 10^0$
	10^4		10^3		10^2		10^1		10^0	

Note: $(\text{any number})^0 = 1$

- Review # of digits in base 10 = 10 - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

Aim: How do we change numbers from base 2 to base 10 and from decimal to binary?

Development:

1. Explain the importance of the binary system with computers.

   . Computers originally worked in base 10. They needed 30 vacuum tubes to represent a number such as 345. Computers that work in binary needed only 9.

Figure 2

Binary

345 expressed in a computer using decimal system and vacuum tubes

0 = off

● = on

10^2	10^1	10^0
●	●	●
●	●	●
●	●	●
0	●	●
0	0	●
0	0	0
0	0	0
0	0	0
0	0	0
0	0	0

---

345 expressed in a computer using binary system and vacuum tubes

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
●	0	●	0	●	●	0	0	●

2. By using base 2 (binary) instead of base 10, computers could be smaller, faster, and have a simpler conversion: circuits are on (1) or off(0). The two digits, 0 and 1, are called "bits" which is a shortening of the words binary digit.

3. Review place value in base system b:

Figure 3

Binary

base exponent:	b^5	b^4	b^3	b^2	b^1	b^0
meaning:	$b \cdot b \cdot b \cdot b \cdot b$	$b \cdot b \cdot b \cdot b$	$b \cdot b \cdot b$	$b \cdot b$	b	1

4. Review number of digits in any system b.

. Number = b, starting at 0 to b-1.
If b > 10, then we start alphabet 10 = A, 11 = B, 12 = C, etc.

5. Binary system summary:

Figure 4

Binary

base exponent:	2^5	2^4	2^3	2^2	2^1	2^0
meaning:	b·b·b·b·b	b·b·b·b	b·b·b	b·b	b	1
value:	32	16	8	4	2	1

Number of digits = 2
0,1

6. Changing from base 2 to base 10:

Figure 5

Binary

Change 1010110_2 to base 10

a) Find the place values in base 2

2^6	2^5	2^4	2^3	2^2	2^1	2^0
64	32	16	8	4	2	1
1	0	1	0	1	1	0

b) Multiply place value times quantity

1 x 64 = 64
0 x 32 = 0
1 x 16 = 16
0 x 8 = 0
1 x 4 = 4
1 x 2 = 2
0 x 1 = 0
Total = 86

7. Changing from base 10 to base 2:

Figure 6

Binary

Change 286 to base 2

a) Write place values in base 2 until you exceed the given number.

256	128	64	32	16	8	4	2	1
1	___	___	___	___	___	___	___	___

b) Subtract the highest place value from the given number. Put a "1" in that place.

286	256	128	64	32	16	8	4	2	1
-256	___	___	___	___	___	___	___	___	___
30									

c) Subtract the highest place value from the remainder. Put a "1" in that place. Put a "0" in any intermediate places.

30	256	128	64	32	16	8	4	2	1
-16	1	0	0	0	___	___	___	___	___
14									

d) Repeat step c until remainder = 0. Fill in unused place values with "0"

256	128	64	32	16	8	4	2	1
1	0	0	0	1	1	1	1	0

14
-8
6
-4
2
-2
0

$286 = 100011110_2$

The subscript indicates the base. When no subscript is written base 10 is understood.

Applications:

1. Change these binary numbers to base 10:

a) 101_2 (5) c) 110110_2 (54) e) 110_2 (6) g) 111101_2 (61)

b) 1011_2 (11) d) 11010_2 (26) f) 101010_2 (42) h) 100001_2 (33)

2. Change these decimal numbers to base 2:

a) 15 (1111_2) c) 56 (111000_2) e) 101 (1100101_2) g) 139 (10001011_2)

b) 32 (100000_2) d) 87 (1010111_2) f) 123 (1111011_2) h) 257 (10000001_2)

Summary:

1. What are the place values of the first five places in the decimal system?

2. What are the digits of the base 10 system?

3. What are the place values of the first five places in the binary system?

4. What are the digits of the base 2 system?

5. How can the place values of the first five places in any base system (b) be represented?

6. How many digits will a base b system have?

7. Why is the binary system used instead of the decimal system in our computers?

Assignnments:

1. Change these binarv numbers to base 10.

a) 11_2 c) 100100_2 e) 100010_2 g) 1001100_2

b) 1101_2 d) 10101_2 f) 111111_2 h) 1011010_2

2. Change these decimal numbers to base 2.

(a) 17 (b) 39 (c) 67 (d) 98
(e) 113 (f) 156 (g) 333 (h) 258

Assignment for More Advanced Students:

1. Develop an algorithm for adding binary numbers. Check your answers by changing the addends and sum to base 10.

Topic 50
HEXADECIMAL/BINARY CONVERSIONS

Performance Objectives: Students will be able to:

- determine the place value and digits in base 16 (hexadecimal)
- review place values and digits in binary, decimal, and any base system
- list two reasons why the hexadecimal system is used by computer programmers
- write hexadecimal numbers in base 10
- write binary numbers in hexadecimal system

Vocabulary: New: hexadecimal
Review: decimal, binary, place value, exponents, base

Motivation:

- Change this decimal number into a number in the binary system: 2619.

- Elicit from the students that it is very hard to check answers when there are so many zeros and ones to read.

- Explain that when you have pages and pages of zeros and ones it becomes really difficult to read a program.

- Review high level languages vs.low level languages (assembly or machine language). Remind students that the high level languages (like BASIC) are faster for programmers while the low level languages are faster for the machine. There will come a time in the future where knowledge of the binary and hexadecimal system may come in handy for those continuing to program computers.

Aim: How do we change base 2 to base 16 and how do we change base 16 to base 10?

Development:

1. Review the place values and digits of base 10, base 2 and base b in general.

Decimal

base exponent:	10^5	10^4	10^3	10^2	10^1	10^0
value:	100,000	10,000	1000	100	10	1

Binary

base exponent:	2^5	2^4	2^3	2^2	2^1	2^0
value:	32	16	8	4	2	1

Base b

base exponent:	b^5	b^4	b^3	b^2	b^1	b^0
meaning:	b·b·b·b·b	b·b·b·b	b·b·b	b·b	b	1

2. Elicit place values from the pupils as powers of 16 and place values of the hexadecimal system. Review prefix "hex" meaning "six" and stem "decimal" meaning "ten" to form "hexadecimal."

base exponent:	16^4	16^3	16^2	16^1	16^0
meaning:	16·16·16·16	16·16·16	16·16	16	1
value:	65,536	4,096	256	16	1

3. Show the relationship between the binary and the hexadecimal system with 4 bits = 1 place value in base 16.

Binary:	2048	1024	512	256	128	64	32	16	8	4	2	1
		256					16				1	
Hexadecimal:												

ex change 110101110110_2 to base 16

a) starting from right, partition each 4 places

1101 0111 0110

b) Convert each set of 4 bits to a number < 16 when number > 9 convert to letter equivalent

(10 = A, 11 = B, 12 = C, 13 = D, 14 = E, 15 = F)

8421	8421	8421
1101	0111	0110
D	7	6

Ans. $110101110110_2 = D76_{hex}$

We use the hexadecimal system because the 0's and 1's of the binary system are difficult to read and because there is an easy conversion from one to the other.

4. Using the place value of base 16, convert numbers from hexadecimal to decimal system:

- Recall place values in base 16.
- Convert digits > 9 to decimal equivalents.
- Multiply each place value by its decimal quantity.
- Add these products to find decimal value of given number.

a) Change $A3B_{hex}$ to decimal notation

256	16	1
A	3	B

No.	x	place value	=	product
10	x	256	=	2560
3	x	16	=	48
11	x	1	=	11
Total			=	2619

b) Change $175F_{hex}$ to decimal notation

4096	256	16	1
1	7	5	F

No.	x	place value	=	product
1	x	4096	=	4096
7	x	256	=	1792
5	x	16	=	90
15	x	1	=	15
	Total		=	5993

Applications:

1. Change these numbers from binary to hexadecimal.

a) 101_2 (5_{hex})

b) 1101_2 (D_{hex})

c) 1011011_2 ($5B_{hex}$)

d) 1010111011_2 ($2BB_{hex}$)

e) 101011001110_2 (ACE_{hex})

f) 1000001_2 (41_{hex})

g) 101111101101_2 (BED_{hex})

h) 10110111011101111_2 ($16EEF_{hex}$)

2. Change these numbers from hexadecimal to decimal.

a) 9_{hex} (9)

b) $5B_{hex}$ (91)

c) $2BC_{hex}$ (700)

d) 67_{hex} (103)

e) BDE_{hex} (3038)

f) ACE_{hex} (2766)

g) $463F_{hex}$ (17983)

h) $23C1_{hex}$ (9153)

Summary:

1. What are the digits of the base 16 system?

2. What are the values of the first four places in the hexadecimal system?

3. Why is knowledge of the hexadecimal system important to computer programmers?

Assignments:

1. Change these binary numbers to base 16.

a) 110_2	e) 1011101_2
b) 1011_2	f) 1000001_2
c) 10101_2	g) 1100011_2
d) 110111_2	h) 110110110111_2

2. Change these hexadecimal numbers to base 10.

a) 23_{hex}	e) $BEAD_{hex}$
b) $9A_{hex}$	f) $1CE8_{hex}$
c) AB_{hex}	g) 3976_{hex}
d) 259_{hex}	h) $FACE_{hex}$

Assignment for More Advanced Students:

1. There are other base systems that are easy to convert from binary. With the eight-bit byte, we could convert every eight bits to a number system. What base would this be? How many digits would it need? What would the first four place values be?

Topic 51
NUMBER BASE SYSTEMS

Performance Objectives: Students will be able to:

- change hexadecimal numbers to binary system
- change decimal numbers to the hexadecimal system
- review changing a binary, decimal, or hexadecimal number to either of the other two bases

Vocabulary: Review: binary, decimal, hexadecimal, place value, bit, base

Motivation:

- What number would the hexadecimal number CAB be in base two? In base 10?

- Review relationship between binary and hexadecimal. (Every four places of binary = 1 place of hexadecimal.)

- Review digits > 9 in base 16 (A = 10, B = 11, C= 12, etc.).

- Review how we write 10, 11, 12 in binary.

CAB_{hex}

$C = 12 = 1100_2$
$A = 10 = 1010_2$
$B = 11 = 1011_2$

C	A	B
1100	1010	1011

$CAB_{hex} = 110010101011_2$

- Convert the binary number to base 10. (3243)

Aim: How can we convert hexadecimal numbers to binary numbers and decimal numbers to hexadecimal numbers?

Development:

1. Review steps 1 - 3 from the motivation and change these hexadecimal numbers to binary:

   - For each place in hexadecimal, four places are needed in binary.

   - Convert any letters to base 10 equivalents.

   - Change digits to base 2, right justifying in the binary places, adding zeros to fill in any blank places to the left of the number.

a) Change $FACE_{hex}$ to binary

$F = 15 = 1111_2$
$A = 10 = 1010_2$
$C = 12 = 1100_2$
$E = 14 = 1110_2$

F	A	C	E
1111	1010	1100	1110

$FACE_{hex} = 1111101011001110_2$

b) Change $D731_{hex}$ to binary

$D = 13 = 1101_2$

D	7	3	1
1101	0111	0011	0001

$D731_{hex} = 1101011100110001_2$

2. To convert decimal numbers to hexadecimal numbers:

(Note: Many teachers allow the use of calculators with this topic to check the arithmetic. However, they do ask the students to show the conversion work since some calculators have built in hex to dec to binary conversion functions.)

- Write place values in base 16 until the place value exceeds the value of the given number.

- Divide the given number by the largest power of 16 which is smaller than the given number.

- If the integer portion of the quotient $>$ 9, convert the number to its corresponding letter (i.e. 10 = A, 11 = B, etc.). Place that digit in the proper place value position.

  (Note: If the integer portion of the quotient $>$ 15, you should be using a higher power of 16.)

- Divide the next highest power of 16 into the remainder and follow the steps above until you reach the units place. Fill in with zeros any place value where division gives you an answer $<$ 1.

Change 13,542 to hexadecimal

4096	256	16	1
3	4	E	6

```
      3
4096/13542
   -12288
     1254

      4
256/1254
    1024
     230

     14
16/230
   -16
     70
   -64
      6
```

Applications:

1. Complete the chart

Decimal	Hexadecimal	Ans.
245	?	($F5_{hex}$)
291	?	(123_{hex})
801	?	(321_{hex})
2783	?	(ADF_{hex})
2914	?	($B62_{hex}$)
4095	?	(FFF_{hex})
4102	?	(1006_{hex})
17198	?	($432E_{hex}$)

2. Complete the chart

BINARY	DECIMAL	HEXADECIMAL
10101_2	?	?
1011011_2	?	?
?	83	?
?	521	?
?	?	$A6_{hex}$
?	?	$2C9_{hex}$

Summary:

1. Why is there an easy conversion from hexadecimal numbers to binary numbers?

2. What steps are followed to convert a number from decimal to hexadecimal?

Assignments:

Complete the chart

BINARY	DECIMAL	HEXADECIMAL
1010_2	?	?
110111_2	?	?
11_2	?	?
11011011110_w	?	?
?	29	?
?	71	?
?	384	?
?	816	?
?	?	D_{hex}
?	?	$3B_{hex}$
?	?	$A47_{hex}$
?	?	$DEAD_{hex}$

Section

7

PROGRAMMING WITHOUT BASIC

Topic 52
COMPILERS

Performance Objectives: Students will be able to:

- understand the function of a compiler
- learn why compiled source code runs faster
- distinguish between interpreters and compilers
- understand the advantages and disadvantages of a compiler

Vocabulary: New: compiled source code, compiler, machine language, interpreter
Review: integrated circuit (I.C.), binary, hexidecimal

Motivation:

- Divide the class into two groups. Instruct the class to find the sum of ten numbers as fast as possible. They are to indicate completion by raising their hands when the problem is solved. The first group is given an alphabet with each letter being assigned a specific value. This group must convert the letters into values, before they can do the addition. The second group is given the numbers directly.

  Example: A=5, B=17, C=34, D=88 etc.

  Group I: Please add A + C + D + B, etc.

  Group II: Please add 5 + 34 + 88 + 17

- Elicit that the additional step required to interpret letters into numbers is a slower process.

Aim: What are compilers and how are they useful in programming?

Development:

1. If all computers essentially work with on and off circuits, how do we communicate with a computer? (We communicate through an interpreter, which is an integrated circuit (I.C. chip) found in the computer's central processing unit (CPU). For example, BASIC is a computer interpreter which converts specific English words into machine language.)

2. What is machine language? (Machine language is the code that computers actually use to control the on and off circuits of the computer. This code can be expressed in binary (base 2) notation or hexidecimal (base 16) notation.)

3. Must an interpreter be built into a computer? (No. Pascal, Fortran, COBOL are examples of different interpreters which can be loaded into the computer's memory from a disk.)

4. What is a compiler? (The compiler is an interpreter which converts a program written in a language such as BASIC or COBOL into machine

language all at once. The user can either save or run this newly compiled source code.)

5. Why do compiled programs run faster than programs written, let's say in BASIC? (As in the motivation, it is obvious that the fewer the translations the faster the execution.)

   . A higher level language such as BASIC can only translate one instruction at a time into a machine language instruction. Whenever the BASIC program repeats an instruction it must once again translate that instruction into machine language.

   . Increased runtime speeds may be 10 to 20 times faster than BASIC programs.

6. What are some of the disadvantages of using a compiler?

   . The compiler program is an additional expense. Fortunately, like many items for the computer, costs are coming down. But it is still expensive.

   . It takes time to compile a program, not to mention the time it takes to learn to use the compiler. Depending on the type of compiler you use, a number of steps are required to complete the process.

   . Compiled programs generally require more computer memory as well as more disk storage space. However, this, too, is becoming less of a disadvantage since computer memory and improved storage devices are becoming cheaper. For many applications, the increased computer processing speed is not necessary since modern computers have faster running times.

   . Compiled programs are not any faster when it comes to I/O (input or output) devices such as printers.

   . Some programs may be too large to be compiled.

Applications:

1. How do we use a compiler?

   . The specific use of a compiler, would, of course, depend on the type of computer and compiler program (there can be several compilers available for a specific computer).

2. A typical application would be one in which a program you have created (source code) is saved to disk. Then the compiler program is booted. The compiler program would then provide a menu which would ask for the name of the program you wish to compile, where in memory the compiled program will reside, and other system requirements.

   . Some compilers work in two passes. The first pass checks to see if the program is valid, checks user inputs, and sets up program parameters. The second pass actually creates the machine language version of your program.

- Many compilers also require that a utility program provided by the compiler be on the same disk as the compiled source code.

Summary:

1. How do we communicate with a computer?

2. What is a compiler?

3. What are some of the advantages of using a compiler?

4. Why do compiled programs run faster?

5. What are some of the disadvantages of using a compiler?

Assignments:

1. List three specific program applications where a compiler would be useful.

2. Discuss three specific program applications where the use of a compiler would not be useful or efficient.

3. Describe how the motivation relates to the use of a compiler.

Assignment for More Advanced Students:

1. Write a program similar to today's motivation which would show how an additional translation or interpretation takes more time. Assign values to each letter of the alphabet. Then, using FOR/NEXT loops, have the program select letters of the alphabet at random (ASC values of 65 to 90) and find the sum of these numbers. Compare this to the speed of simply adding a given number of randomly chosen numbers.

Topic 53
THEORY OF LANGUAGES — MACHINE LANGUAGE

Performance Objectives: Students will be able to:

- state the three main parts of a computer
- describe a microprocessor
- describe the different registers
- explain the purpose of a bus
- state the advantages and disadvantages of machine language

Vocabulary: New: machine language, microprocessor, register, accumulator, index registers, data bus, address bus
Review: binary numbers

Motivation:

- How can a programmer program so that the computer responds most efficiently? Every program is translated into a set of binary numbers. A machine language program contains instructions with only 0's and 1's. If a program in machine language is entered into the memory of a computer, then the computer can respond to the program directly.

Aim: How does the computer respond to machine language instructions?

Development:

1. A computer consists of the following three main parts: (a) logic circuits or directing intelligence of the computer. The logic circuits continually repeat the instruction execution cycle of reading an instruction, interpreting it, and executing it; (b) memory to store programs, data, and intermediate results; (c) input/output circuits communicate with the keyboard, speaker, and expansion slots.

2. The microprocessor or central processing unit (CPU) is a single integrated circuit which controls the logic, that is instruction execution cycle. The microprocessor for the Apple and Commodore machines is the 6502, for the TRS 80 it is the Z80, and for the IBM PC it is the 8088. The microprocessor controls the registers, the locations where numbers are stored and manipulated by programs. The accumulator is the most commonly used register. In the 6502, the accumulator is an 8-bit storage element with connections. Storage and movement of data and arithmetic operations are often done by the accumulator. An 8-bit number is a binary number with 8 digits, that is a number from 0 to 255. The accumulator uses the arithmetic logical unit to perform calculations.

3. Other registers are index registers, called X and Y, which are used to point to items in a list of data or instructions. Another register is the program counter, which holds the memory address of the next program instruction. In the 6502, there is a 16-bit register and it can address any of the 65,536 locations or bytes. The stack pointer register is used for temporary storage of information. The status register indicates the results of different calculations, such as whether a number is positive or negative, and is used to control program branching.

4. There are communication lines into and out of the CPU. The group of lines which transmits one 8-bit byte as a unit is called a bus. There are 2 main types of buses: (a) data bus transmits data, usually between the memory and CPU; (b) address bus selects the location from which data is to be transferred.

5. Let us assume that the microprocessor (6502) is ready to execute an instruction. The program counter contains the address of the instruction. The microprocessor uses the address bus to call that address to place its information on the data bus. The instruction can be represented as an 8-bit signal, where 1 could represent high voltage and 0 represent low voltage. An example of an instruction is 10110010, usually written as 1011 0010 in order to make it easier to translate into hexadecimal. This instruction is stored in an instruction register. The instruction is then decoded by the microprocessor into different actions using electronic logic, involving decision and memory circuits. The microprocessor then uses the program counter to determine the address of the next instruction.

6. Although machine language programs are executed more quickly than any other programs, the disadvantages of machine language programming are that the programs are difficult to write and debug.

Applications:

1. Convert 8-bit binary numbers into hexadecimal.

2. Convert hexadecimal numbers into decimal numbers and decimal numbers into hexadecimal numbers.

Summary:

1. What are the main parts of a computer?

2. What is the microprocessor?

3. What are the purposes of the different types of registers?

4. What are the different types of buses?

5. What are the advantages and disadvantages of machine language programming?

Assignments:

1. Find the instructions in machine language for your microprocessor for loading data into the accumulator, storing data in memory, and adding numbers.

2. Find a list of OP codes for the different machine language instructions for your microprocessor.

3. Find examples of small machine language programs to perform calculations using FOR/NEXT loops.

Assignment for More Advanced Students:

1. Write a research paper explaining electronic logic, or switching logic. Your paper should include a discussion of decision and memory circuits, gates, and flip flops.

Topic 54
THEORY OF LANGUAGES — ASSEMBLY LANGUAGE

Performance Objectives: Students will be able to:

- describe the advantages and disadvantages of assembly language programming
- describe the advantages and disadvantages of higher level language programming

Vocabulary: New: hexadecimal loader, assembler, mneumonics, higher level languages, compiler
Review: hexadecimal, binary

Motivation:

- What are the sixteen single-digit hexadecimal numbers? Hexadecimal is a convenient bridge between the binary numbers of machine language and the decimal numbers. Each group of four binary number digits corresponds to one hexadecimal number. It is easier to remember 16 hexadecimal digits than the 256 binary digits of machine language. A language which utilizes hexadecimal numbers and instruction codes in English is assembly language.

Aim: What are assembly language and higher level languages?

Development:

1. Since the computer only understands binary instruction codes, there is a program called a hexadecimal loader which converts hexadecimal numbers to binary numbers.

2. However, writing in hexadecimal is still difficult. Therefore, instruction code names, called mnemonics, can be assigned. Examples of mnemonics are ADD, AND, OR. An assembly language program consists of mnemonics and hexadecimal numbers. An assembler is a program which translates a source program in assembly language into an object program which is machine language. Assemblers require more memory and are more expensive than hexadecimal loaders, but it is easier to write in assembly language than to write with only hexadecimal numbers.

3. Disadvantages of assembly language programming are: (a) the programmer must have detailed knowledge of the microcomputer (the registers, addressing methods); (b) assembly language programs can not be transferred between different microcomputers; (c) it is still time-consuming and difficult to write programs in assembly language.

4. Advantages of assembly language programming are: (a) it encourages one to understand the internal workings of the computer; (b) the programs may be executed more quickly than with a language such as BASIC. This is an advantage in game programs, for example.

5. Higher level languages, such as BASIC, allow programmers to communicate with the computer in a language closer to English. A compiler is a program that translates the higher level language into machine language. Advantages of higher level languages over assembly or machine language are: (a) writing and debugging are easier; (b) the programmer does not

need to understand about registers and methods of addressing for the computer; (c) programs are more portable from computer to computer.

6. Disadvantages of higher level languages over assembly or machine language are: (a) compilers are slow to execute; (b) compilers take up a large amount of memory.

Application:

1. Compare the speed of assembly language and higher level language programs to perform similar operations.

Summary:

1. What is a hexadecimal loader?

2. What is assembly language?

3. What are advantages and disadvantages of assembly language?

4. What are advantages and disadvantages of higher level languages?

Assignments:

1. Find the assembly language instruction codes for your microcomputer.

2. Find examples of small assembly language programs to perform calculations using FOR/NEXT loops.

Topic 55
THEORY OF LANGUAGES — HIGHER LEVEL LANGUAGES

Performance Objectives: Students will be able to:

- distinguish between general purpose and special purpose languages
- distinguish between compiled and interpreted languages
- explain the features of some higher level languages

Vocabulary: New: general purpose, special purpose, interpreted, compiled
Review: structured programming

Motivation:

- Which higher level computer language would you use for each of the following programs?

  - business
  - scientific
  - mathematical
  - educational
  - graphic

Aim: What are the reasons for using the different higher level languages?

Development:

1. Higher level languages can be divided into the following types:

   - general vs. special purpose

   - interpreted vs. compiled

   - structured vs. unstructured

   - interactive vs. noninteractive

2. General purpose languages include Cobol, BASIC, ALGOL, LISP, APL, PL/I, and Fortran. Special purpose languages include PILOT for computer-aided instruction, PROLOG for logic programming, and C and FORTH for systems programming.

3. A compiler translates a program into machine language, which is then executed by the computer. An interpreter performs the program line by line. Although a compiled language executes faster, an interpreted language is better for beginners because program revisions can be made immediately. Fortran and Cobol are usually compiled while BASIC may be compiled or interpreted.

4. With newer computer languages such as Pascal it is difficult to write poorly structured programs. With older languages such as Fortran or BASIC, the programmer can write well structured or poorly structured programs.

5. With interactive languages, such as BASIC, APL, LOGO, and PILOT, the

programmer can more easily edit and detect errors.

6. Following are descriptions of some of the more popular computer languages:

   - Cobol is a language for business applications. Its strengths are its data handling, matrix features, and file handling.

   - Fortran is often used for mathematical and scientific applications. Its string handling capabilities are limited.

   - BASIC is supposed to be easy to learn. It comes in compiled or interpreted versions. There are now extensive business, educational, and scientific BASIC software packages.

   - Pascal is a structured language which is being used more and more in schools.

   - LISP is used for artificial intelligence because it allows the manipulation of lists and data for pattern recognition, algebraic manipulation, and language processing.

   - PILOT is used for the development of drills and dialogues for computer aided instruction.

   - C is a language which has the strengths of assembly language but is also more easily transferrable between different computers.

   - LOGO is often taught to elementary school students. Its features are recursion (the ability of the program to call itself) and a turtle for graphics.

Application:

1. Investigate the features of FORTH and Smalltalk.

Summary:

1. What is the difference between general purpose and special purpose languages?

2. What is the difference between structured and nonstructured languages?

3. What are the advantages of interactive languages?

4. Describe features of the following languages: Cobol, Fortran, BASIC, Pascal, LISP, PILOT, C, and LOGO.

Assignments:

1. Describe features of the following languages: AIDA, PL/I, APL, and ALGOL.

2. Classify computer languages into general purpose and specific, structured and nonstructured, interactive and noninteractive.

Assignment for More Advanced Students:

1. What are future trends for higher level computer languages?

Topic 56
AN INTRODUCTION TO LOGO
PRIMITIVE COMMANDS AND TURTLE GRAPHICS

Performance Objectives: Students will be able to:

- learn the rudiments of using the LOGO language
- become familiar with the use of the LOGO system primitives
- create simple drawings using the LOGO commands
- learn how to edit their text from within the LOGO system
- access the different screen modes from within the LOGO system
- use the REPEAT command to simplify their instructions to the TURTLE

Vocabulary: New: LOGO, Terrapin LOGO, LOGO system primitive, REPEAT
Review: high resolution graphics, hires, fullscreen, splitscreen, textscreen

Motivation:

- Inform students that they are being introduced to the LOGO language through the programming area commonly referred to as "turtle geometry." LOGO'S graphic commands are only one part of this very powerful computer language. LOGO includes commands for arithmetical operations and list processing. There is also a line editor included in the language that takes the drudgery out of screen editing. Provisions are made for the addition of the right and left bracket keys for computers that cannot access these keys from the keyboard.

- Introduce the class to the turtle (draw a facsimile on the board). The LOGO turtle is a "cybernetic animal" that lives on the computer screen and that is controlled by the computer. She/he/it (your choice) responds to LOGO commands. Most of these commands direct the turtle to move to selected screen positions. As the turtle moves it leaves a visible trail. This enables us to create drawings on the video screen. We can instruct the turtle to move without leaving this trail and, if color monitors are available, we can select the color that we want to draw with. Our turtle is intelligent and will learn procedures that you, the programmer, will teach it. These aspects of LOGO are so simple that they have been effectively taught to very young children. LOGO, however, is also taught at the graduate level.

- Assure your students that by the end of the period they will be creating drawings using the LOGO language.

Aim: What are some of the commands used by the TERRAPIN LOGO language, and how are they used to created drawings on the high resolution screen?

Development:

1. The following list of commands should be copied and handed out to each student.

********** LOGO SYSTEM PRIMITIVES **********

GRAPHICS COMMAND	ABBREVIATION	FUNCTION
FORWARD	FD	Turtle will move in the direction in which it is pointing.
BACK	BK	Turtle moves in the opposite direction.
RIGHT	RT	Turtle moves right (in a clockwise direction) an indicated number of degrees.
LEFT	LT	Turtle moves left (in a counter clockwise direction) an indicated number of degrees.
HIDETURTLE	HT	Hides the turtle from view.
SHOWTURTLE	ST	Turtle reappears on video screen.
DRAW	(NONE)	Shows a clear high resolution screen, with turtle in its initial position.
NODRAW	ND	Gives the text screen with the cursor at the top leftmost position.
CLEARSCREEN	CS	Clears the hires screen (does not change the current position of the cursor).
HOME	(none)	Brings turtle back to its initial position (does not clear screen).
PENUP	PU	Turtle will not draw line as it moves.
PENDOWN	PD	Turtle will draw line as it moves.
WRAP	(none)	(default mode) Causes subsequent turtle commands to "wraparound" in response to attempts to move the turtle past the screen boundaries.
NOWRAP	(none)	Causes LOGO to signal the error "turtle out of bounds" in response to any attempt to move the turtle past the screen boundaries.
BACKGROUND	BG	Sets background color.

****

2. How do we begin?

. The LOGO language is not built into the Apple computer as the BASIC language is. We must first teach our computer the language by loading it into the Apple's memory via the disk drive. Does this change the way we communicate with the computer? Absolutely! The cursor now appears preceded by a question mark. This indicates to us that LOGO is awaiting a command. By typing DRAW or HOME the high resolution screen is displayed with the turtle in the center (the default mode is mixed graphics:splitscreen). The text window at the bottom provides the familiar four lines of text. This is the area we will use for entering our LOGO commands.

3. Are there any changes in the way we enter text?

. A few. If an incorrect letter is typed, it cannot be deleted by simply retyping over the incorrect letter. The escape key is used as our eraser as we type, or we can position our cursor over the incorrect entry and delete it using CTRL-D. When you use the escape key it will erase the key to the left of the cursor. Complete lines can be erased with the CTRL-K (kill).

4. Are there changes in the screen size?

. Naturally. The initial position of the turtle is at the center of the screen. Think of this point as the origin. Our screen size is 280 by 240. If you think of the turtle as being positioned at (0,0) then there are 140 steps to the right and left, and 120 steps above and below. Since the Apple computer provides us with 24 screen lines, we can think of the height of one line of text as being 10 steps. Thus the four lines of text at the bottom hide the lowest 40 steps. If you wish to see all of your picture in splitscreen mode, then do not advance the turtle more than 80 steps below its initial position. The default of the LOGO DRAW mode is to use wraparound rather than declare an out-of-bounds error. Changing these modes is best discussed after the students have some experience using the primitive commands.

5. How do we use these commands?

. We are going to use the abbreviations to inform the turtle of our wishes. To draw a vertical line from the turtle's initial position to the top of the screen we type FD 120. Note the space between the command and the number of steps. If you forget the space then LOGO will respond with, "There is no procedure named FD120." BK 200 will give us a vertical line ending at the top of the text window. HOME will return the turtle to its initial position and the line will still appear on the screen. Let us attempt to draw a horizontal line and complete a set of axes. Our turtle will rotate 360 degrees enabling us to draw lines in every direction. It is important to realize that once we start rotating the turtle each subsequent rotation is dependent on the current position of the turtle. Since our last command was HOME the current position of the turtle is 0 degrees. To draw a horizontal line from the center to the right of the screen we must rotate our turtle 90 degrees to the right (clockwise). The command will be RT 90. The turtle is now positioned to draw the line. Elicit FD 140 will draw the line. BK 280 will complete the axes. An alternate procedure for drawing this horizontal line could have been accomplished by LT 90 FD 140 BK 280; or RT 270 FD 140 BK 280. Bring the turtle back to its

initial position and clear the screen. (HOME CS) It is important that the students receive some guidance with relative turtle positions and rotation. To eliminate confusion it is helpful to present a simple drawing. A square or an equilateral triangle are about the easiest. For the triangle RT 120 FD 50 RT 120 FD 50 RT 120 FD 50. For the square FD 50 RT 90 FD 50 RT 90 FD 50 RT 90 FD 50. Just one or two simple procedures is all an average group will need to get them started. It is also helpful to demonstrate the rotation procedure on the chalkboard.

6. Is it necessary to use a separate line for each command?

   . No. As long as the line does not exceed 255 characters it will be accepted. However if they wish they can press return after each command. If the student is not familiar with the text window it could be a bit confusing when their text starts to disappear.

7. Is there a way to access fullscreen and textscreen from within LOGO?

   . Yes. CTRL-T gives text mode. CTRL-F gives fullscreen graphics mode. CTRL-S gives splitscreen mode.

   . Students will need time to experiment with the LOGO Primitives presented thus far. Several minutes of "hands on" is recommended at this time. The aim is to establish the need for commands that will simplify lengthy instructions, speed up the drawing procedure and let the turtle do the bulk of the work.

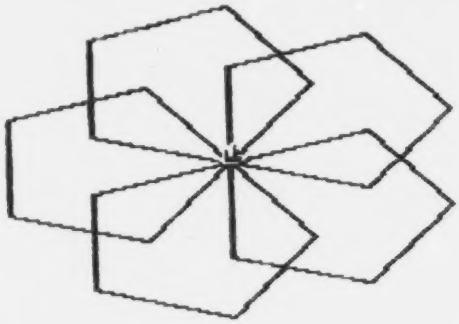
8. Introduce the REPEAT command. This command is accessed by writing out the word REPEAT followed by a number which represents the amount of repetitions desired. The procedure that we want repeated is then enclosed in brackets. For computers that do not have these keys the open and closed square brackets are typed on the Apple keyboard as SHIFT-N and SHIFT-M, respectively. We say the REPEAT command takes two inputs. The first input is a number and the second input is a list of commands enclosed in square brackets. The result of this command is to repeat the set of commands in the list, the designated number of times. The square that we drew earlier in the lesson can be drawn using the REPEAT command.

   . REPEAT 4[FD 50 RT 90]

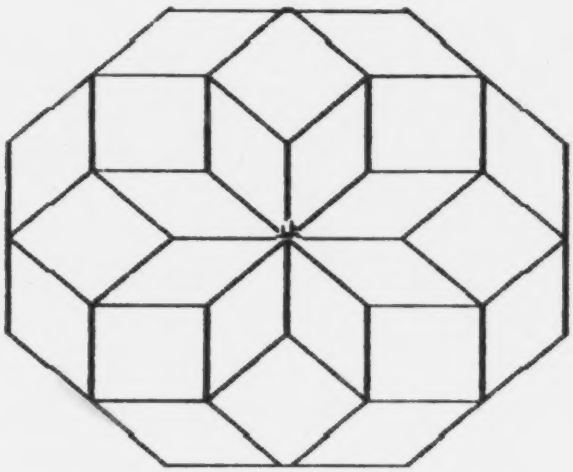
     - The equilateral triangle can be drawn with REPEAT 3[FD 50 RT 120]. Draw two triangles with REPEAT 2[REPEAT 3[FD 50 RT 120] PU FD 60 PD]. Note the way the REPEATS are nested. Students should be familiar with this type of nesting from their knowledge of BASIC FOR/NEXT loops. Some very beautiful designs can be created by drawing the regular polygons in overlapping patterns. Remind students of the formula for finding the size of the exterior angle of a regular polygon of N sides. (360/N) A pentagon would have an exterior angle of 72 degrees. After we draw the initial pentagon we rotate our turtle 72 degrees and draw a second pentagon. If we do this five time we complete a rotation of 360 degrees.

     - Suggested drawings:

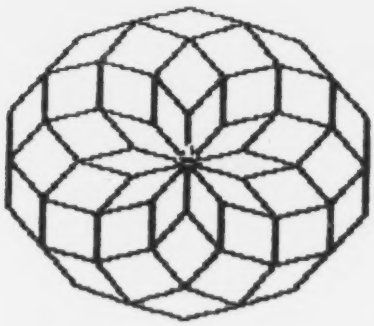
Five pentagons: REPEAT 5[REPEAT 5[FD 50 RT 72]RT 72]



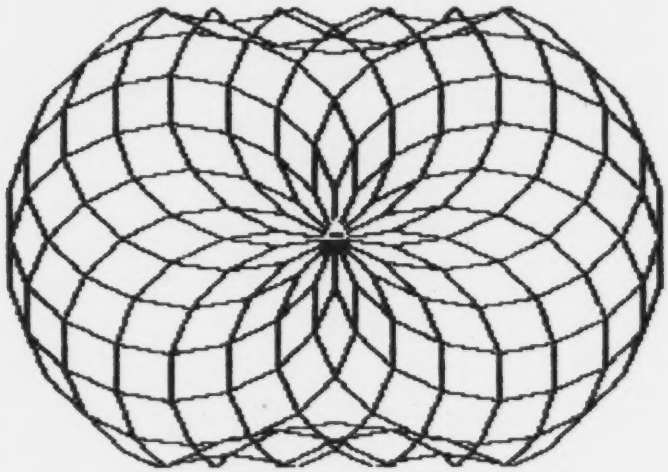
Eight octagons: REPEAT 8[REPEAT 8[FD 40 RT 45]RT 45]



Ten decagons: REPEAT 10[REPEAT 10[FD 20 RT 36]RT 36]

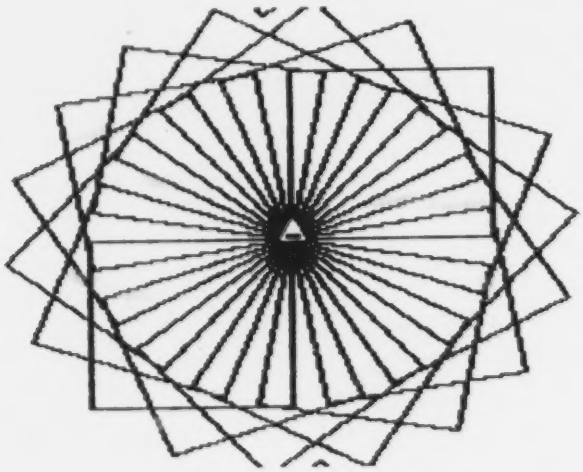


Eighteen 18-gons: REPEAT 18[REPEAT 18[FD 20 RT 20]RT 20]



- It might be helpful to point out at this time the result of increasing the number of sides of the polygon. We will be using this principal in subsequent lessons to draw circles and arcs. It is not necessary to draw the same number of figures as the number of sides of the polygon. If a symmetrical figure is desired we need only to complete a 360 degree rotation. Try drawing 18 squares. Since 360/18 is 20 we will rotate our turtle 20 degrees after drawing each square.

  REPEAT 18[REPEAT 4[FD 70 RT 90]RT 20]



Applications:

1. Direct the turtle to draw a rhombus with two perpendicular diagonals. Begin the drawing by rotating the turtle 45 degrees. Make use of the REPEAT command to draw the rhombus. Use a 90 degree angle for the

rhombus. Remind class of the formula used to find the length of the diagonal. Approximate the square root of 2 as 1.41. Discuss the way LOGO accepts arithmetical operations as inputs for its primitive commands.

Acceptable program:

```
RT 45 REPEAT 4[FD 50 RT 90]
LT 135 BK 50 * 1.41 LT 45
FD 50 RT 135 FD 50 * 1.41
```

2. Use the program you just wrote to draw the figure in the upper left hand corner of the video screen.

   Acceptable addition:

   PU FD 85 LT 90 FD 105 RT 90 PD (insert program here).

   Now draw the same figure in the upper right hand corner so that there will be two identical drawings on the top of the screen.

   Acceptable addition:

   (Insert first addition and program) RT 90 PU FD 150 PD

   (Insert program here)

3. Use the square and equilateral triangle we drew earlier to draw a large square with a small triangle drawn at each corner. Make use of nested repeats. First draw the triangles outside the square, then draw the triangles inside the square. Vary the sizes of each figure to achieve interesting effects.

   Square with outside triangles:

   REPEAT 4[REPEAT 3[RT 120 FD 25] FD 80 RT 90]

   Square with inside triangles:

   REPEAT 4[REPEAT 3[FD 50 RT 120] FD 80 RT 90]

4. Use nested REPEATs to draw six hexagons. Determine that after each drawing the turtle should be rotated 60 degrees.

   Six hexagons:

   REPEAT 6[REPEAT 6[FD 50 RT 60] RT 60]

5. Use the REPEAT command to draw a rectangle. Add a vertical line and change the rectangle into a flag. Use nested REPEATs to draw three flags.

   Three flags:

```
LT 90 PU FD 100 RT 90 PD
REPEAT 3 [REPEAT 2[FD 30 RT 90 FD 50 RT 90] BK 80
PU RT 90 FD 80 LT 90 FD 80 PD]
```

Summary:

1. What do we do first when we attempt to draw with the aid of the LOGO turtle?

2. What do we mean by the phrase "turtle geometry?"

3. What is the size of the LOGO drawing screen?

4. What letters will the following lists of commands draw? (Assume we begin with the turtle in standard position.)

   FD 50 RT 90 FD 25 BK 50 (T)

   BK 50 RT 90 FD 25 (L)

   FD 50 RT 90 FD 25 BK 25 RT 90 FD 20 LT 90 FD 25 (F)

5. Why is it necessary to leave a space between a command and an input?

6. Are there LOGO commands that take no inputs?

7. What keystrokes accomplish the following commands?

   . FULLSCREEN

   . TEXTSCREEN

   . SPLITSCREEN

8. Name two commands that clear the drawing screen? (CS and DRAW)

9. Can we see pictures that are drawn after the HT command? (Yes, it is sometimes quicker to draw without displaying the turtle.)

10. Which of the following lists of commands will draw 5 pentagons?

```
REPEAT 5[ REPEAT 5[ FD 30 RT 72 RT 72]
REPEAT 5[ RT 72 REPEAT 5[ FD 30 RT 72]]
REPEAT 5[ REPEAT 5[FD 30 RT 72]] RT 72
REPEAT 5[REPEAT5[FD 30 RT 72] RT 72]
```

   . (The first two will work, the third will redraw the same pentagon five times, and the fourth will not work because there is no space between the second repeat and the input 5.)

Assignments:

1. Write a list of commands that will draw a pinwheel by repeating a rectangle eight times. Make sure the turtle is rotated correctly after each rectangle is drawn. Be sure to nest your repeats correctly.

2. A duodecagon is a twelve sided polygon. Write a list of commands that will draw twelve regular duodecagons and return the turtle to its initial position when the drawing is completed.

3. Use the REPEAT command to outline the screen with squares. This works nicely with a square of side twenty, but you can use any size square you wish.

Assignments for More Advanced Students:

1. Draw a flag with all horizontal stripes.

2. Draw a flag divided into three vertical stripes.

3. Draw a flag similar to the American flag using dots for stars.

Answers to Assignments:

1. REPEAT 8[REPEAT 2[RT 90 FD 80 RT 90 FD 20] RT 45]

2. REPEAT 12[REPEAT 12[FD 20 RT 30] RT 30]

3.
```
PU FD 80 LT 90 FD 140 RT 90 PD
REPEAT 9[REPEAT 4[ FD 20 RT 90] PU BK 20 PD]
REPEAT 13[REPEAT 4[FD 20 RT 90] PU RT 90 FD 20 LT 90 PD]
PU HOME
PU FD 80 LT 90 FD 140 RT 90 PD
REPEAT 13[REPEAT 4[FD 20 RT 90] PU RT 90 FD 20 LT 90 PD]
PU HOME
PU FD 80 RT 90 FD 100 LT 90 PD
REPEAT 9[REPEAT 4[ FD 20 RT 90] PU BK 20 PD]
HT
```

Answers to Assignments for More Advanced Students:

1.
```
PU BK 50 PD FD 100
REPEAT 2[FD 50 RT 90 FD 100 RT 90]
REPEAT 9[REPEAT 2[FD 5 RT 90 FD 100 RT 90]FD 5]
```

2.
```
PU BK 50 PD FD 100
REPEAT 2[FD 50 RT 90 FD 100 RT 90]
RT 90 FD 100/3 LT 90 FD 50
RT 90 FD 100/3 RT 90 FD 50
```

3.
```
PU BK 50 PD FD 100 REPEAT 2[FD 50 RT 90 FD 100 RT 90]
REPEAT 5[REPEAT 2[FD 5 RT 90 FD 100 RT 90]FD 5]
REPEAT 5[REPEAT 2[FD 5 RT 90 FD 60 RT 90] FD 5]
BK 5 LT 90
REPEAT 4[REPEAT 5[PU FD 5 PD FD 2]
PU BK 35 RT 90 BK 5 LT 90 PD]
RT 90]
REPEAT 5[REPEAT 2[FD 5 RT 90 FD 100 RT 90]FD 5]
REPEAT 5[REPEAT 2[FD 5 RT 90 FD 60 RT 90] FD 5]
BK 5 LT 90
REPEAT
```

Topic 57
EDITOR PROCEDURES IN LOGO

Performance Objectives: Students will be able to:

- use the LOGO editor to write procedures
- formulate new procedures by editing previously written ones
- write simple procedures that draw geometric shapes
- inform LOGO to accept input through the use of the colon
- learn the technique of having one procedure call another procedure

Vocabulary: New: procedure, editor
Review: primitive, repeat, input

Motivation:

- In our last lesson we learned how to draw with LOGO by using its PRIMITIVES. Today we will teach the turtle to respond to words that we create. By the end of this lesson I will not only type ⟨forward, back, right and left⟩ but when I type square, LOGO will draw a square, or a triangle or whatever I teach it to draw when I say square. I could teach our turtle to draw a square when I say GOO. The turtle is very obedient and will learn whatever I teach it, without any back talk. Three months from now, however, I might forget what GOO does, but I will assume SQUARE will draw a square. Writing procedures to draw simple shapes and including these procedures in other procedures that we write is, in effect, the way we program in LOGO.

Aim: How do we teach the turtle to respond to new commands that we create?

Development:

1. LOGO employs some very useful editing techniques. CTRL-G is used to stop the execution of a drawing. If we meant to type REPEAT 3 and we typed REPEAT 33 by mistake, it is not necessary to wait out this procedure. By typing CTRL-G the execution of the procedure stops immediately. Another technique for halting the execution of a procedure is to type CTRL-W. This will cause the drawing to stop and wait for you to examine it or do whatever you choose. When you wish to continue the procedure LOGO will accept any keystroke to continue executing the procedure. Think of CTRL-W as a pause, or a "wait a minute" command. Therefore we will use CTRL-G to stop permanently and CTRL-W to stop temporarily.

   - CTRL-P will cause the previously input line to be retyped. This is helpful when you are testing out repeat procedures. Access the line that was just executed with CTRL-P and then press RETURN. This will cause the previous line to be re-executed. If an error occurred in typing, for example, you may have typed BA instead of BK, and LOGO responded with an error message, press CTRL-P and the input line will reappear with the cursor positioned at the end. You may edit this line and immediately press RETURN. It is not necessary to reposition the cursor at the end of the line as you would do in BASIC. Repeated pressing of CTRL-P will cause the line to appear again and again. If you are using this procedure remember to leave a space before each new

line appears. Try drawing a set of axes with the following procedure.

- Enter the following list.

  FD 100 BK 100 RT 90 RETURN
  CTRL-P ⟨space⟩ CTRL-P ⟨space⟩ CTRL-P RETURN.

- It is important to note that if a line contains a REPEAT command it cannot be accessed again with the CTRL-P.

- Remembering one line is nice but it is hardly a powerful enough tool for our purposes. Today we are going to learn how to write programs in LOGO. Anytime we want LOGO to remember a procedure we must write it in the editor and give it a name. We indicate that we are about to write a procedure and would like to enter the editor by using the word TO. For example if we wish to draw a house we might say TO HOUSE. Next we might say TO WINDO followed by TO ROOF or TO GARDEN. As soon as the TO is encountered, the high resolution screen will disappear and you will find yourself in the editor. Before we use the editor, however, it is necessary that we learn some editing techniques. A few simple keystrokes will make writing and editing your procedures very simple.

The following list of editing keystrokes should be copied and handed out to the students.

********** EDITING **********

To enter the LOGO editor:
Type TO ⟨program name⟩, or
Type EDIT, or
Type EDIT ⟨program name⟩

You must enter the LOGO editor to:

Write a program, or
Edit a previously entered program.

While in the editor use the following keystrokes for cursor control.

KEYSTROKE	FUNCTION
CTRL-A	Moves the cursor to the beginning of the current line.
CTRL-B	Moves the cursor back one screenful of text. This is used to scan long programs that fill many screens.
CTRL-C	Used to exit the editor when you are satisfied with your program and ready to have it processed.
CTRL-D	Deletes the character at the current position of the cursor.
CTRL-E	Moves the cursor to the end of the current line.
CTRL-F	Moves the cursor forward one screenful of text. Used for

```
***** TO START:
PU LT 90 FD 50 RT 90 FD 60 PD
END

***** TO MOVED: (down move)
PU RT 90 FD 50 RT 90 FD 20 RT 90 PD
END

***** TO MOVER:  (right move)
PU FD 20 LT 90 BK 15 PD
END
```

. Students should be made aware that by experimenting and then defining these procedures they are breaking the large program down to simple procedures. They are now ready to draw three windows across and three windows down, with a door centered at the bottom.

```
***** TO FILL:
START
REPEAT 3 [REPEAT 2 [WINDO MOVER] WINDO PU BK 80 MOVED]
PU FD 30 RT 90 FD 35 RT 180 PD  (position door)
WINDO
END
```

. Now all that remains is for them to put everything together to draw the house. They have to draw the outline send the turtle back home, fill in the windows and door and finish by hiding the turtle.

```
***** TO HOUSE:
OUTLINE
PU HOME
FILL
HT
END
```

4. LOGO has its own input command. It is the colon and we call it dots. Let us teach LOGO to draw a square and reserve the right to decide the length of its side at a later time. We will call the variable length "side." We indicate that this is a variable by preceding it with the colon.

```
***** TO SQUARE :SIDE
REPEAT 4 [FD :SIDE RT 90]
END
```

. Note that every occurrence of the word "side" must be preceded by dots. If you enter SQUARE, LOGO will respond with SQUARE NEEDS MORE INPUTS. We must indicate the length of the side each time we want the turtle to draw a square. SQUARE 50 or SQUARE 10 is the correct way to call for this procedure.

5. Many designs can be created with small variations on the same figure. Return to the simple square figure and add two diagonals. Review with students the method of finding the length of the hypotenuse. Approximate the square root of 2 as 1.41.

long programs that fill many screens.

CTRL-G — Exits the editor without processing the edited text. When not in edit mode this keystroke is used to stop the turtle's drawing procedure.

CTRL-K — Deletes all characters on the current line to the right of the cursor. The end of a line is reached when a carriage return is encountered.

CTRL-L — Scrolls the text so the line containing the cursor is in the center of the editing screen. Used when writing programs that take more than one screen.

CTRL-N — Moves the cursor down to the next line. (Remember a line ends when the computer encounters a carriage return.)

CTRL-O — Opens a new line at the cursor's position. This keystroke is used for inserting text in a previously written program or program segment.

CTRL-P — Moves the cursor to the previous line. This keystroke has a very powerful use in DRAW or NODRAW mode. It retrieves the previously input line so that it can be re-executed or edited before re-execution.

3. Review editing procedures with students. Now we are ready to draw our house. Type TO OUTLINE. Discuss the format of the editor with the students. Agree to make the house 150 by 130. Position the outline in the center of the screen.

```
***** TO OUTLINE
PU BK 45 RT 90 FD 65 PD (positions cursor to begin drawing)
REPEAT 2[LT 90 FD 150 LT 90 FD 130] (draws rectangle)
END (necessary)
```

. Press CTRL-C and LOGO will accept the program. The next screen will say OUTLINE DEFINED. Now every time you type OUTLINE the turtle will follow the procedure you just wrote. Now draw a window. Agree to make the window 20 by 30. We can then rotate the window 90 degrees and use the same drawing for a door. Type TO WINDO and find yourself in the editor. (Note: The word window is a primitive in some versions of LOGO and therefore we are calling our routine WINDO instead of WINDOW.)

```
***** TO WINDO:
REPEAT 2[FD 30 RT 90 FD 20 RT 90]
FD 15 RT 90 FD 20
END
```

. After pressing CTRL-C you will see the message WINDO DEFINED. Let students experiment with OUTLINE and WINDO, discuss correct positioning of the cursor before each drawing and establish the need for a starting position, a DOWN MOVE and a RIGHT MOVE. When naming these procedures we must be careful not to use any words that LOGO uses to name its primitives.

```
***** TO BEGIN :SIZE
REPEAT 4 [ FD :SIZE RT 90 ] (draws square)
RT 45 FD :SIZE * 1.41 (draws first diagonal)
LT 135 FD :SIZE RT 45 BK :SIZE * 1.41 (draws second diagonal)
END
```

- Try repeating this figure eight times in succession. Name the procedure A.

```
***** TO A :SIZE
REPEAT 8 [ BEGIN :SIZE ]
END
```

- By changing the position of the turtle at the end of our original drawing we can achieve another result. Add to our BEGIN procedure the step RT 90 just before the END. Next position the turtle so that our drawing does not go out of bounds. Finally, repeat the procedure that we named A.

```
***** TO B :SIZE
PU FD 1.5 * :SIZE PD (positions cursor)
A :SIZE
END
```

- The third change that we are going to make will be to move the turtle away from our original picture after it is drawn. We will then repeat this procedure the same eight times. Add to our BEGIN procedure the following steps just before END and after RT 90.

```
PU BK :SIZE PD
```

- Now it is again necessary to position our turtle so that our picture will not go out of the screen's boundaries.

```
***** TO C :SIZE
PU FD :SIZE + 10 LT 90 BK :SIZE RT 90 PD (position cursor)
A :SIZE
END
```

- For our fourth and final change we will return to "BEGIN" and change the RT 90 (which we inserted when we made our first change) to RT 180. This will have the effect of moving the turtle away from our original diagram at a different angle. This time it is not necessary to reposition our turtle at the beginning. A simple HOME statement is all we need.

  ***** A :SIZE will now yield a new result.

6. Circles can easily be drawn with LOGO. One way of drawing a circle is to draw a polygon of 360 sides. Try REPEAT 360 [FD 1 RT 1]. You can speed up the procedure with HT.

```
***** TO CIRCLE
HT
REPEAT 360[FD 1 RT 1]
END
```

This limits us to only one size circle. We can make use of the input feature of LOGO to draw a circle of any size.

```
***** TO CIRCLE1 :SIZE
HT
REPEAT 360[FD :SIZE RT 1]
END
```

. This procedure, while being more useful, is still cumbersome. We usually think of determining a circle's size by the length of its radius. In the above procedure the circumference is determined by multiplying the size by 360. We think of C = 2(Pi)r. Hence 360(SIZE) = 2(Pi)r. Solving this equation for the radius yields r = SIZE(180)/Pi. We can incorporate this in our procedure and then call for a circle by the size of its radius.

```
***** TO CIRCLE2 :RADIUS
HT
CIRCLE1 :RADIUS * 3.14159/180
END
```

. Now that we can draw circles it is useful to have a procedure that draws arcs. We need an arc of 180 degrees for a semicircle and others of varying degrees to draw petals on flowers. In effect we need to draw a part of a circle which is determined by any number of degrees.

```
***** TO PARTCIRCLE :SIZE :DEGREES
HT
REPEAT :DEGREES [FD :SIZE RT 1]
END
```

. Now, we can draw a semicircle with the command PARTCIRCLE 1 180. An examination of this procedure reveals that the arc is drawn toward the right. Simply changing RT 1 to LT 1 will draw the arc in the opposite direction.

```
***** TO ARCLEFT :SIZE :DEGREE
HT
REPEAT :DEGREES [FD :SIZE LT 1]
END
```

. It might to clearer now to rename PARTCIRCLE, ARCRIGHT.

Applications:

1. Write a procedure that will draw a rectangle of any length and any width.

```
***** TO RECTANGLE :L :W
REPEAT 2 [ FD :L RT 90 FD :W RT 90]
END
```

. Test your procedure with RECTANGLE 30 50, ETC.

. Edit this procedure so that you can draw a pinwheel of any size.

```
***** TO PINWHEEL :L :W
REPEAT 8[REPEAT 2[FD :L RT 90 FD :W RT 90] RT 45]
END
```

. Test procedure with PINWHEEL 30 80

. If we change our final angle to 30 degrees from 45 degrees would it be necessary to change the number of repeats? (yes to 12)

2. Draw a petal by using the ARCRIGHT procedure and drawing an arc of 60 degrees.

```
***** TO PETAL
HT
REPEAT 2[ARCRIGHT 1 60 RT 120]
END
```

. Make use of the PETAL to draw a flower.

```
***** TO FLOWER
HT
REPEAT 6[PETAL RT 60]
END
```

. How would you draw a flower with eight petals? (Change the number of repeats to 8 and the angle to 45 degrees).

3. Draw concentric circles. Write a procedure that positions the turtle for drawing. Use the length of the radius to move the turtle away from its initial position. Make use of the circle procedure we wrote that depends on the length of the radius.

```
***** TO CONCENTRIC :RADIUS
HT
PU LT 90 FD :RADIUS RT 90 PD
CIRCLE2 :RADIUS
PU HOME PD
END
```

. Test your procedure with CONCENTRIC 60 and then CONCENTRIC 40, etc.

4. Birds can be represented by drawing their wings in flight. Experiment by adding some birds of various sizes to the sky above our house. Use the ARCLEFT and ARCRIGHT procedures we wrote to create the bird.

```
***** TO BIRD :WING :DEGREE
HT
ARCRIGHT :WING :DEGREE
PU HOME PD
ARCLEFT :WING :DEGREE
END
```

. Test the procedure with BIRD .6 70, or BIRD .4 80

5. Write a procedure that will draw an equilateral triangle of any size.

Combine this procedure with the RECTANGLE procedure we wrote earlier to draw a rocket.

```
***** TO TRIANGLE :S
REPEAT 3[FD :S RT 120]
END

***** TO ROCKET :L :W
RECTANGLE :L :W
FD :L RT 180 TRIANGLE :L
LT 90 FD :W LT 45 FD 20 BK 20
RT 45
REPEAT 2[RT 90 FD :L/3 LT 90 FD 20 BK 20]
RT 90 FD :L/3 LT 45 FD 20
HT
END
```

- Test your procedure with ROCKET 30 70. What would be the result of inserting a RT 90 at the beginning of the procedure? (The rocket would be drawn vertically instead of horizontally.)

Summary:

1. Where are our LOGO procedures written?
2. How can we move the cursor to the previous line in edit mode?
3. Does this keystroke have the same effect in draw mode?
4. What is the effect of changing the name of our procedure in edit mode? (a new procedure is defined)
5. How do we signal LOGO that we want a procedure to accept keyboard input?
6. How do we stop the turtle from drawing a picture?
7. What is the result of drawing a polygon of 360 sides?
8. How can we draw a semicircle?
9. Do all procedures need additional inputs?
10. How do LOGO procedures call other procedures?

Assignments:

1. Modify the FLOWER procedure we wrote to draw a rose of 12 petals.
2. Combine the CONCENTRIC procedure we wrote with the flower you just drew to draw the flower inside the circle.
3. Draw 3 vertical ROCKETS in three different sizes on one screen.

Assignment for More Advanced Students:

1. Draw a portion of the sky. Include a cloud, a rocket, and some birds of

different size.

Possible answers to Exercises:

1.
```
TO ROSE
REPEAT 12[PETAL RT 30]
END
```

2.
```
TO CIRCLEROSE
HT
CONCENTRIC 60
PU HOME PD
ROSE
END
```

3.
```
TO ROCKET3
RT 90
ROCKET 30 70
PU HOME FD 20 RT 90 FD 90 PD
ROCKET 20 90
PU HOME FD 40 RT 90 BK 90 PD
ROCKET 25 120
END
```

Topic 58
PERIPHERAL COMMANDS AND ADVANCED PROGRAMMING IN LOGO

Performance Objectives: Students will be able to:

- save the procedures they write as LOGO DISK FILES
- make hardcopies of their procedures as well as their pictures
- learn primitive commands that will enable them to position the turtle more easily
- change PENCOLOR to black for the purpose of erasing unwanted parts of a drawing
- understand the way LOGO handles conditional commands
- experiment with the way LOGO handles recursion

Vocabulary: New: SETHEADING, HEADING, SETXY, READPICT, SAVEPICT, OUTDEV, PRINTOUT, POTS, recursion
Review: hardcopy, primitive, IF-THEN-ELSE, STOP, READ, SAVE, endless loop

Motivation:

- After LOGO is started up, direct class to call for a procedure they wrote at the previous lesson. After it is established that the procedures were not saved, produce the disk on which you saved the workspace of the previous lesson. Inform students that they will soon learn to save their pictures as well as their procedures.

- A discussion of the homework should point to the need for a simpler way of positioning the turtle to begin successive drawings. Other commands that students should miss from BASIC are the conditional commands. Assure the class that these commands do exist in LOGO and that they will soon be quite satisfied at the ease with which they can direct the turtle.

Aim: What are some of the additional primitives that LOGO uses to communicate with the peripherals and to write more advanced procedures?

Development:

1. As procedures are written in LOGO they are saved in the area of LOGO'S memory called "the workspace." They will remain there until the computer is shut off or you erase them with the ERASE command. The workspace is not endless and occasionally LOGO will respond to your attempts to write a new procedure <WORKSPACE IS FILLED PLEASE ERASE SOMETHING>. It's a good idea to get into the habit of erasing unwanted procedures as they lose their usefulness. This command is abbreviated ER and is followed by the name of a procedure.

2. At the end of a work session you will want to save useful procedures. You do this with the SAVE command. The way we signal LOGO that we wish to communicate with the disk drive is to precede the filename that we wish to save with a quotation mark. One quotation mark is all that is needed. We do not close the quote. An example of this command is SAVE "NEWFILES. The space before the quote is required as SAVE is a LOGO

primitive and must be followed by a space. When you save a file at the end of a session it is important to note that you are saving the complete workspace. You may have written ten procedures at that session, they will all be saved in the file named NEWFILES.

3. To see the names of the procedures that are currently in the workspace you can direct LOGO to print out the titles. This is accomplished with the POTS command. POTS is an abbreviation for PRINTOUT TITLES. When LOGO encounters this command it will give you a list of all the procedures that are currently in the workspace. Remember that if you had previously erased a procedure it is no longer in the workspace.

4. If you are interested in seeing what you wrote in any given procedure, then you can make use of the LOGO PRINTOUT command. This primitive is abbreviated PO and simply typing PO SQUARE will printout the procedure on the textscreen. If you happen to be in DRAW mode at the time you will have to press CTRL-T to see the commands in the procedure. Remember this printout is on the textscreen and not in the EDIT BOX and therefore cannot be changed.

5. Sometimes it's useful to have a hardcopy of a printout. To access the printer the command is OUTDEV (SLOT#). If the printer is accessed by PR#1 in BASIC then use OUTDEV 1 in LOGO. If you want a printout of all your procedures then PO ALL is all you need type. Make sure you include the space as PRINTOUT is a LOGO primitive. If you simply want a hardcopy of the procedure you named SPIRAL then type OUTDEV 1 PO SPIRAL. To disengage the printer type OUTDEV 0, or press CTRL-SHIFT-M.

6. To obtain a copy of a picture that you created is slightly more complicated. This depends on your printer and the way it is interfaced to your computer. You can, however, save your picture as a LOGO binary file and then consult your manuals to determine the way it can be dumped by your printer. SAVEPICT is the command that LOGO uses to save the hires screen. Since this is a command to the disk drive, the picture name must be preceded by the single quote. SAVEPICT "HOUSE is an example of the way LOGO saves a picture.

7. Now that we have saved pictures and procedures on our disk we want to be able to load them back into our computer's memory at a later time. The command CATALOG will give us a listing of the files that are currently on the disk. That will not be hard to remember as it is the same in BASIC. To load a file from the disk we use the command READ. READ "NEWFILES will load the set of procedures that we saved earlier in the lesson. READPICT "HOUSE will bring back, onto the hires screen, the picture that we saved. When we read our catalog we see that LOGO had added suffixes to our files. A file of procedures such as NEWFILES appears in our catalog as NEWFILES.LOGO. HOUSE appears in our catalog as HOUSE.PICT. These suffixes are useful in determining the type of file we saved but they need not be used when we READ them back.

8. LOGO also includes commands that delete files from the disk. ERASEFILE "NEWFILES and ERASEPICT "HOUSE are the commands that will delete from the disk the files that we saved earlier.

9. Although we have made no attempt to treat color in these lessons, the need may arise to erase parts of drawings. This can be accomplished by

drawing over any unwanted lines using the color black. In LOGO pencolors are accessed by number. While drawing on a black background the color white is 1 the color black is 0. PENCOLOR is the LOGO primitive used to direct the turtle to draw in specific colors. The following procedure is useful in demonstrating how drawing in black can be used to erase.

```
***** TO COLORTEST
REPEAT 4[PC 1 FD 120 PC 0 BK 120 RT 90]
END
```

10. Remind students that they can recall an input line with CTRL-P. If this is used in conjunction with the change in pencolor then erasing becomes a very simple procedure. Assume you have just drawn a design and wish to insert the letter "N" in the center. You position your cursor and type FD 50 RT 135 FD 70 LT 135 FD 50 RETURN.

    . You are not pleased with the result. Recall line with CTRL-P, move cursor to beginning of line, insert PC 0, press RETURN and the "N" will be erased leaving the design untouched. Repeat this procedure, change pencolor to 1 and 70 to 75 and achieve the desired results.

11. Two commands that make our job of positioning the cursor easier are SETXY and SETH (an abbreviation for set the heading of the turtle). The commands follow our usual interpretation of the axes with our turtles default position at (0)(0). Parentheses are used to enclose the X and Y coordinates and we give X first which avoids confusion. The default heading of the turtle is 0 degrees thus SETH 90 is equivalent to RT 90 from the turtle's initial position. SETH 270 is the same as LT 90 from HOME. Remind students that the screen is 280 by 240. Drawing on the very ends of the screen is not recommended. A procedure that outlines the screen is:

```
***** TO OUTLINE
FULLSCREEN
PU SETXY (-139) (-119) PD
REPEAT 2 [FD 239 RT 90 FD 278 RT 90]
END
```

    . Caution students that the primitives RT AND LT are relative to the current position of the turtle while SETH is not. We hope to make this clearer in the next set of drawings.

12. The picture of the house that we drew at our last lesson looks very bare. We are going to attempt to do some landscaping. We begin by drawing a LEAF, then a BRANCH, then a BUSH, and finally a MOVE. This will enable us to draw a GARDEN that contains a series of BUSHes.

```
***** TO LEAF
LT 45 FD 10 BK 10 RT 90 FD 10 BK 10 LT 45
END

***** TO BRANCH
REPEAT 2 [FD 15 LEAF] FD 10 BK 40
END
```

```
***** TO BUSH
LT 60 REPEAT 7 [BRANCH RT 20] lt 80
END

***** TO MOVE
SETH 90 PU FD 65 SETH 0 PD
END

***** TO GARDEN
PU SETH 0 SETXY (-100)(-80) PD
REPEAT 4[BUSH MOVE]
HT
END
```

- The garden drawing may now be added to the HOUSE for an improved picture.

13. Unlike the BASIC language we are familiar with, LOGO programs can make use of recursion. While we do not provide any lengthy explanation at this point, students familiar with programming in the Pascal language will be aware of the recursive nature of the following programs. For the others simply define recursion as the word we use for procedures that call upon themselves. For a more formal definition, try: A recursive procedure is a procedure that repeats itself, with each repetition based in part on the results of the previous one. This has the effect of reducing a complex problem to a series of progressively simpler problems until the solution is reached.

- Make use of the arc procedures developed in the previous lesson to draw an infinity sign.

```
***** TO INFINITY
PU HOME PD
ARCRIGHT .5 120
ARCLEFT .5 120
PU HOME RT 120 PD
ARCLEFT .5 120
ARCRIGHT .5 120
END
```

- Tell students that we are now going to insert INFINITY after the last ARCRIGHT and just before the END. The result will be to set up an endless loop which will cause the turtle to redraw the same figure in the same place over and over again. The drawing will have to be stopped by pressing CTRL-G. Inform students that although this may appear to be quite useless, the fact that the procedure named INFINITY has INFINITY as one of its commands, will greatly enhance their ability to program in LOGO.

14. Remind students of the way they controlled their loops in BASIC. In a FOR/NEXT loop they specified in advance the number of times they wanted a loop to occur. This could be compared to LOGO's REPEAT command. Other loops were terminated by using a counter in conjunction with a conditional statement. LOGO too, has conditional statements. IF/THEN/ELSE and STOP are LOGO primitives. These commands are used the same way they are used in BASIC. For example, IF :SIDE = 0 THEN STOP will terminate a procedure when the condition :SIDE = 0 is met.

- Return to the CONCENTRIC procedure written in the last lesson. Write a procedure that will draw 10 concentric circles by having the procedure call upon itself.

```
***** TO CONCENTRIC1 :RADIUS
HT PU HOME LT 90 FD :RADIUS RT 90 PD
CIRCLE2 :RADIUS
IF :RADIUS = 0 THEN STOP
CONCENTRIC1 :RADIUS - 10
END
```

- Walk through this procedure with the class. Notice that each time a circle is drawn the radius is diminished by 10. LOGO remembers the size of the radius and when it reaches 0 the procedure is terminated. Test the procedure with CONCENTRIC1 100

Applications:

1. Write a general polygon program where we input the length of the side, the number of degrees in the exterior angle, and the number of sides in the polygon.

```
***** TO POLY :LENGTH :ANGLE :SIDES
FD :LENGTH RT :ANGLE
IF :SIDES = 0 THEN STOP
POLY :LENGTH :ANGLE :SIDES - 1
END
```

- A walk through the procedure will indicate that we enter a number for the number of sides that is one less than the number of sides we require. Thus POLY 100 120 2 draws a triangle and POLY 100 90 3 draws a rectangle. Experimenting with this program yields some very beautiful designs.

2. Use the BUSH procedure to draw a tree.

```
***** TO TREE
FD 50
BUSH
BK 50
END
```

- Now draw a FOREST. Begin by positioning the turtle and then provide for drawing any number of trees.

```
***** TO START
FULLSCREEN
PU SETXY (-100)(-100) PD
END
```

- We are going to draw a forest and we want our trees to overlap more than our bushes did. It is necessary to write a new MOVE.

```
***** TO MOVE
SETH 90 PU FD 50 SETH 0 PD
END
```

BEST COPY AVAILABLE

```
***** TO FOREST :DENSITY
START
REPEAT :DENSITY [TREE MOVE]
END
```

3. Draw birds of various sizes that appear to be flying high in the sky. We will use the ARCLEFT and ARCRIGHT procedures. The birds in the foreground will be the largest ones. We will reduce the size of the bird as it flies higher in the sky. Make the starting position a variable so that we can reuse the procedure in different diagrams. Experiment with the length of the wing.

```
***** TO BIRDSKY :X :Y :SIZE
SETH 0 PU SETXY (:X) (:Y) PD
ARCRIGHT :SIZE 70
PU SETH 0 SETXY (:X)(:Y) PD
ARCLEFT :SIZE 70 PU
IF :SIZE < .1 THEN STOP
BIRDSKY (:X + 30) (:Y + 20) :SIZE - .1
END
```

. Test your procedure with BIRDSKY 10 30 .4

4. Draw a representation of the sun. Make a large circle centered on the splitscreen with rays emanating from its circumference. Agree to space the rays every 22.5 degrees.

```
***** TO SUNRAY :SUN :RAY
PD FD 25 BK 25 SETH :SUN
ARCRIGHT 1 22.5 SETH :RAY
IF RAY = 360 THEN STOP
SUNRAY :SUN + 22.5 :RAY + 22.5
END

***** TO SUNRAY1
PU HOME SETXY (0) (70)
SUNRAY :90 22.5
END
```

Summary:

1. Is it necessary to save our procedures after we write each one?

2. Do we save our procedures with the same command as we use to save our pictures?

3. What is the difference between the command ERASE and the command ERASEFILE?

4. How do we load previously saved procedures into the LOGO workspace?

5. What command(s) do we use to obtain a hardcopy of all the procedures that we wrote at a given session?

6. Is there a way to erase a portion of our drawing without clearing the screen?

7. What are the coordinates of the upper left hand corner of our screen?

8. Assume our turtle is in its default position. If I wish to send it diagonally up to the left corner with the SETH # FD command, what number must be used for the heading?

9. What do we mean by a conditional command?

10. What is a recursive procedure?

11. How does LOGO react to the colon, the single quote?

Assignments:

1. Use the BEGIN procedure that we developed in our previous lesson to draw a new picture.

2. Modify a sun so that the rays point inward instead of outward. Fill the space in the center with a five-pointed star.

3. Use the POLY procedure to draw two 36-pointed stars on the same screen. One star should be made with an angle of 130 degrees and the other with an angle of 110 degrees. The length of the side of each must be the same. We used length 100 in the accompanying diagram. Draw a rectangular outline around the figures. How do you account for the difference in their size?

Assignment for More Advanced Students:

1. Draw a diagram by modifying and combining SUNRAY, BIRDSKY, and FOREST.

Possible answers to Assignments:

1.
```
TO BEGIN :SIZE
REPEAT 4 [FD :SIZE RT 90]
RT 45 FD :SIZE * 1.41
LT 45 BACK :SIZE
LT 45 FORWARD :SIZE * 1.41
END
```

```
TO DESIGN
PU SETXY (80) (0) PD
REPEAT 8 [BEGIN 40]
PU SETXY (10) (0)
PD BEGIN 40
END
```

2. First modify SUNRAY
```
TO SUNRAY :SUN :RAY
PD BK 30 FD 30 SETH :SUN
ARCRIGHT 22.5 SETH :RAY
IF :RAY = 360 THEN STOP
SUNRAY :SUN + 22.5 :RAY + 22.5
END
```

```
TO STARWHEEL
SUNRAY1
PU SETXY (-5) (-5)
PD POLY 40 144 4
END
```

3.
```
TO POLYTWO
PU SETXY (-100) (-100) PD
POLY 100 110 35
PU SETXY (40) (-10)
PD POLY 100 130 35
PU SETXY (-130) (-30) SETH 90 PD
REPEAT 2 [FD 255 LT 90 FD 140 LT 90]
HT
END
```

EXTRA.
```
TO PARTSUN :A :B
 PU SETH :B PD
 FD 30 BK 30 SETH :A
 ARCRIGHT 1 19
 IF :A = 240 THEN SETH :B + 10 FD 30 STOP
 PARTSUN :A + 20 :B + 10
END
TO START
 PU SETXY (-100) (120)
 PARTSUN :A :B
END
TO PICTURE
 START 160 90
 PU HOME PD
 BIRDSKY 20 20 .4
 SETH 0 FOREST 5
END

END
TO START
 PU SETXY (-100) (120)
 PARTSUN :A :B
END
TO PICTURE
 START 160 90
 PU HOME PD
 BIRDSKY 20 2
```

Topic 59

AN INTRODUCTION TO PASCAL

Performance Objectives: Students will be able to.

- state the overall format of a Pascal program
- explain the purposes of structured programming
- identify the simple Pascal data types

Vocabulary: New: structured programming, documentation, functions, procedures
Review: algorithms, data types, ASCII codes, compiled

Motivation:

- If you had to write a long, complicated program, what would you perceive to be the advantages of using the BASIC computer language?

- What would you perceive to be disadvantages of using the BASIC computer language? Another language that might be easier to program complicated programs is Pascal.

Aim: What is the overall format of the Pascal computer language?

Development:

1. Pascal, developed by Niklaus Wirth of Switzerland, is a structured language. This means that programs are divided into modules, that is separate sections each with a specific function. It is easier to understand, to document, and to debug programs which are written with a structured language. Documentation explains the purposes, how the program was constructed, and how the program can be changed.

2. In order to set up a structured program, one often uses algorithms, which are sequences of steps needed to solve a problem. It can be as simple as a one-step formula or as complicated as a sort. The problem and steps needed to solve it are clearly enumerated. A problem is broken down into smaller and smaller subproblems, which results in greater refinement of the problem. With the use of algorithms and a language which encourages structured programming, such as Pascal, there is no need to use the two-step process of using flowcharts as a language to communicate with people and then the computer language to communicate with the machine.

3. Variables must be identified at the beginning of a program. Functions and procedures are modules that are called by the main program to serve a specific purpose. Remarks, which are not processed, are indicated with a (* and *). Separate sections of the program have BEGIN and END statements. In general, the punctuation at the end of a line is a ;
The overall format for a Pascal program is:

```
PROGRAM NAME;
(* DECLARATION OF VARIABLES *)
TYPE (* THE USER MAY DEFINE NEW DATA TYPES HERE *)
VAR
     VARIABLE1: DATATYPE1;
     VARIABLE2: DATATYPE2;
```

```
BEGIN
    (* FUNCTIONS, PROCEDURES *)
END;
BEGIN
    (* MAIN PART OF PROGRAM *)
END.
```

4. The different simple data types are:

   . Integer - contains no decimal or comma

   . Reals - the range is larger for a real than an integer

   . Character - any symbol available on the computer

   . Boolean - a Boolean variable can be either true or false

     - Examples are:

       NEEDSTOBETESTED:=FALSE
       GOODSTUDENT:=TRUE

   . String or Packed arrays - up to 255 characters

     (Note: Variables NEEDSTOBETESTED and GOODSTUDENT are assigned values with a := instead of an =.)

   . User defined - just as Boolean represents true or false, a user defined variable can represent any value. For example: TYPE SUBJECTS = (MATH,CHEM,BIO,ENGLISH,SPANISH). Variables can then represent part or all of the type SUBJECTS.

5. Operations with the variables include:

   . Arithmetic operations: +, -, DIV (for integers only), /.

   . Boolean operations: and, or, not.

   . Ordering operations: to determine the predecessor and/or successor in an ordering, to relate characters to their ASCII codes.

6. Methods of inputing and outputing data include:

   . Assignment statements - the notation is := instead of =.

     Examples:

     X:=10

     NAME:='JONES'

   . READ,READLN - With a READ, no values are skipped over. If it is only necessary to read the beginning data on a line, one uses a READLN. The READLN performs a carriage return so that the remaining data (if any) is not read. For example, assume data (entered after the program is compiled) is as follows:

```
10 20
30
```

The following will result in X being 10 and Y being 20.

```
READ (X);
READ (Y);
```

The following will result in X being 10 and Y being 30:

```
READLN(X);
READLN(Y);
```

(Note: The ; is used to separate lines in a Pascal program.)

- WRITE,WRITELN - with the WRITE statement, the variables are output adjacent to each other. WRITELN performs a carriage return after the variables are output. Consider the following:

```
X:=10;
Y:=20;
WRITE (X);
WRITE (Y);
```

The output is:

```
1020
```

If instead you have the following:

```
WRITELN(X);
WRITELN(Y);
```

The output is:

```
10
20
```

Other examples are:

```
WRITE ('GOOD');
WRITE ('MORNING,');
WRITE ('STEVE');
```

the results are:

```
GOOD MORNING, STEVE
```

whereas the results of:

```
WRITE ('GOOD');
WRITELN ('MORNING,');
WRITELN ('STEVE');
```

would be:

```
GOOD MORNING,
STEVE
```

Since the WRITELN statement could be used to generate blank lines, the following statement:

```
WRITELN (X,Y,Z)
```

gives the same result as

```
WRITE (X);
WRITE (Y);
WRITE (Z);
WRITELN
```

Summary:

1. What is meant by structured programming?

2. What is meant by modules?

3. Define documentation.

4. What are the purposes of functions and procedures?

5. Describe the overall format of a Pascal program.

6. What are the different data types?

7. What are the different methods for input and output?

Assignment:

1. Describe differences between BASIC and Pascal for the following topics:

   . Overall format of program

   . Data types

   . Operations with data types

Topic 60
CONTROL STRUCTURES IN PASCAL

Performance Objectives: Students will be able to:

- identify the different control structures
- state a reason for using each type of control structure

Vocabulary: New: control structures, WHILE loop, FOR loop, IF...THEN, IF...THEN...ELSE, CASE
Review: Boolean expression

Motivation:

- What are the methods in BASIC which allow the programmer to branch out of sequence? (Methods used to repeat steps, that is form loops, or to make decisions are called control structures.)

- What is a purpose of each of these methods?

Aim: What are the different control structures in Pascal?

Development:

1. Consider a program to calculate the average salary for a company with an unspecified number of employees. With Pascal, there is a control structure, the WHILE loop, in which a loop is performed as long as a condition, that is Boolean expression, is true. For example:

```
WHILE SALARY > 0 DO BEGIN
    (* READ SALARY *)
    (* CALCULATE SUM,COUNT OF
    EMPLOYEES *)
END;
AVERAGE:=SUM/COUNT;
```

   - As long as the salary is greater than 0, the loop from BEGIN to END will be performed. To exit the loop, the last item in the data should be 0 or negative. The syntax for the WHILE loop is:

```
WHILE (Boolean expression) DO
    BEGIN
    (* THE LOOP *)
    END;
```

2. Let us look at a progam to calculate the average salary for a company with 1000 employees. The control structure in Pascal, which is similar to BASIC, is the FOR loop. For example:

```
FOR COUNT:=1 TO 1000 DO
    BEGIN
        (*READ SALARY*)
        (*CALCULATE SUM*)
    END;
    AVERAGE:=SUM/1000;
```

- The loop will be performed 1000 times. The syntax for the FOR loop is:

```
FOR (VARIABLE):=(INITIAL VALUE)
  TO (FINAL VALUE) DO
      BEGIN
      (*THE LOOP*)
      END;
```

- Note that if the final value is less than the initial value, the word DO is replaced by the word DOWNTO.

- Unlike BASIC, in Pascal the initial and final values need not only be numbers, but can be characters. For example:

```
FOR LETTER:='A' TO 'Z' DO
   BEGIN
       (*WRITE EACH LETTER*)
   END;
```

Another example:

```
FOR LETTER:='Z' TO 'A' DOWNTO
   BEGIN
       (*WRITE EACH LETTER*)
   END;
```

3. The IF...THEN statement from BASIC is expanded in Pascal to allow for multiple alternatives. The new control structure is IF...THEN...ELSE. An example of a program to arrange two letters, LETTER1 and LETTER2, in alphabetical order is:

```
IF LETTER1>LETTER2 THEN
     (* WRITE LETTER2 FIRST *)
 ELSE
     (* WRITE LETTER1 FIRST *)
```

The form, in which each condition is either true or false (a Boolean expression) is:

```
IF CONDITION1 THEN
     (* FIRST ALTERNATIVE *)
ELSE
     (* SECOND ALTERNATIVE *)
```

The following program segment uses all three types of loops. The segment reads 20 integers, one per line of input. If an integer is negative or zero, it will be printed. If it is positive, the computer will print a number of asterisks on one line equal in length to the read number.

```
BEGIN
   FOR COUNT:= 1 TO 20 DO
    BEGIN
     (*READ THE NEXT NUMBER*)
```

```
      READLN (NUMBER);
      IF NUMBER <=0 THEN
       (*WRITE NUMBER IN NUMERICAL FORM*)
       WRITE (NUMBER)
      ELSE
        BEGIN(*WRITE THAT NUMBER OF ASTERISKS*)
        WRITE(' ');
        ASTERISK:=0;
      END;(*ELSE*)
      WHILE ASTERISK < NUMBER DO
       BEGIN
        WRITE ('*');
        ASTERISK: = ASTERISK+1
      END(*WHILE*)
    END;(*FOR LOOP*)
     (*START ANOTHER LINE OF OUTPUT*)
     WRITELN
   END;
```

Applications:

1. Using a WHILE loop, write a program to compute students averages.

2. Using a FOR loop, write a program to print the sum of the squares of even integers from 2 to 50.

Summary:

1. What is a control structure?

2. What are the different methods of control structures?

3. What is the syntax for a WHILE loop?

4. What is the syntax for a FOR loop?

5. What is the syntax for IF...THEN...ELSE?

Assignments:

1. Using control structures, write programs to:

   . Find the largest of 3 integers.

   . Print the interest accumulated for each account at a bank.

   . Convert numeric grades to letter grades.

   . Print the sum of the squares of multiples of 5 from 25 to 625.

Topic 61
DATA STRUCTURES IN PASCAL

Performance Objectives: Students will be able to:

- identify the different data structures
- state a reason for using each type of data structure

Vocabulary: New: data structures, sets, files, records, union, intersection
Review: array

Motivation:

- Assume you are the programmer for Grand Central Station. What types of information or data would you need to collect and what kind of programs would you have to write to manage it?

Aim: How do we create in Pascal variables, called data structures, which allow us to store several types of data at once?

Development:

1. With data structures, a variable can represent the simple data types (integer, real, character, Boolean). One of the strengths of Pascal is that it allows the programmer to represent complicated relationships among data.

2. Suppose you are the programmer for Grand Central Station. If you wanted to have different simple types of data stored you might use the RECORD data structure. Each simple data type in a RECORD is called a field. For example, at the beginning of a program you might declare a record as follows:

```
TYPE TRAIN=RECORD
          NAME:STRING;
          DESTINATION:STRING;
          DEPARTURETIME:REAL;
          ARRIVALTIME:REAL;
          GATE:CHAR;
          TRAINFULL:BOOLEAN
      END;
```

   - Note that there are several simple data types (string, real, char, Boolean) in the above RECORD. There are six fields.

3. For a collection of the same types of data, you might use an ARRAY. You may have an array of simple data types (such as an array of integer values) or an array of records. For example, if you had 1000 trains, you might declare a record as was done previously and then declare:

```
TYPE STATION=ARRAY[1..1000]
     OF TRAIN;
VAR TRAINSTATION:STATION;
```

   - Therefore, there are 1000 records, ranging from TRAINSTATION [1] through

TRAINSTATION[1000]. Note that you may combine the different data structures.

4. To store large amounts of data on a permanent basis, you would use a FILE data structure. You might want to store rules and regulations or fare schedules. For example, you might declare a sequential file of data (characters) as follows:

```
TYPE STORAGE=FILE OF CHAR;
 VAR PERMANENT:STORAGE;
```

5. If you wanted to use set operations, such as union, intersection, difference, or inclusion, for sets of different routes, you would use the SET data structure. An example of a declaration is:

```
TYPE CITY=[NY,LA,PHIL,BOSTON,
      CLEVELAND];
      SCHEDULE=SET OF CITY;
 VAR TRAVELTONY,TRAVELTOLA,
     TRAVELTOPHIL,
     TRAVELTOBOSTON,
     TRAVELTOCLEVELAND:SCHEDULE;
```

. The following are examples of operations:

```
TRAVELTONY:=[BOSTON,CLEVELAND];
  (* assignment statements *);
TRAVELTOLA:=[PHIL,BOSTON];
TRAVELTONY+TRAVELTOLA:=
  [BOSTON,CLEVELAND,PHIL];
  (*the union operation
  includes elements in either
  set*);
TRAVELTONY*TRAVELTOLA:=[BOSTON];
  (*the intersection operation
  includes only those elements
  that are in both sets*);
```

Consider another example. Let us say you are the programmer for a bank. A program you write might have the following TYPE statement:

```
TYPE ACCOUNT=RECORD
 NAME:STRING
 ADDRESS:STRING;
 ID:INTEGER;
 DEPOSIT:REAL;
 WITHDRAWAL:REAL;
 BALANCE:REAL;
 OVERDRAWN:BOOLEAN;
END;(*RECORD*)
(*AN ARRAY OF ACCOUNT RECORDS*)
NAMES=ARRAY[1..1000] OF ACCOUNT;
BANK=FILE OF RECORD;
TRANSFERS=[BROOKLYN,BRONX,MANHATTAN,QUEENS,STATEN ISLAND];
TRANSACTION=SET OF TRANSFERS;
```

Application:

1. Assume that you are the chief programmer for a university. Describe how you would use each of the data structures.

Summary:

1. What is the purpose of each of each of the following data structures?

   . Records

   . Arrays

   . Sets

   . Files

Assignments:

1. Assume you are chief programmer for a large company. Explain the purpose of and declare each of the following data structures:

   . Records

   . Arrays

   . Files

   . Sets

2. Explain the difference between BASIC and Pascal with regard to data types and data structures.

Section

8

SOME USEFUL MATHEMATICS TOPICS

Topic 62
STATISTICS — MEASURES OF CENTRAL TENDENCY

Performance Objectives: The students will be able to:

- write a program to determine the arithmetic mean of a set of numbers
- write a program to determine the median of a set of numbers
- decide when the arithmetic mean or median is the appropriate measure of central tendency
- write a program to determine the range of a set of numbers
- write a program to determine the variance and standard deviation of a set of numbers

Vocabulary: New: statistics, measures of central tendency, arithmetic mean, median, measures of dispersion, range, variance, standard deviation, mode

Motivation:

- One community has citizens with the following assets:

ASSETS	# OF PEOPLE
$1,000,000,000	1
$1,000	99

- Another community has citizens with the following assets:

ASSETS	# OF PEOPLE
$10,000	100

- Which community has the richer citizens? (The collection, organization, and analysis of data such as that above is done by statisticians. Statistics has many important applications in research and industry.)

Aim: In how many different ways can the average measure of data be analyzed?

Development:

1. The motivation problem shows that the average, that is, the arithmetic mean, of a set of numbers may not provide an adequate interpretation of the data. The arithmetic mean of the assets of the first community is greater than that of the second community, but one might consider the citizens of the second community to be richer. There appears to be a need to have more than one type of average measure or measure of central tendency

2. There are three measures of central tendency: a) the arithmetic mean or average is defined as the sum of the numbers divided by the number of numbers; b) the median is the middle number when the numbers are sorted; c) the mode is the number occurring most frequently. There can be more than one mode.

3. The following program segment can be used to determine an arithmetic mean of 100 numbers:

```
10 DIM A(100)
20 SUM=0
30 FOR I=1 TO 100
40 INPUT A(I)
50 SUM=SUM+A(I)
60 NEXT I
70 WRITE 'AVERAGE IS ';SUM/100
```

4. In order to determine the median of a set of numbers, the numbers must first be sorted (for example, by a selection sort or bubble sort). If the number of elements is odd, then the median is the middle element. For example, if there are 99 elements, then the 50th element is the median. If the number of elements is even, then the median is the arithmetic mean of the two elements around the middle. For example, if there are 100 elements, then the median equals the sum of the 49th and 50th elements divided by 2.

5. After the numbers are sorted, a program segment to find the median of numbers might be as follows:

```
100 IF N/2<>INT(N/2) THEN I=INT(N/2) +1: PRINT MEDIAN IS ";A(I):GOTO
     120: REM N IS ODD
110 I=N/2: PRINT  "MEDIAN IS "; (A(I)+A(I+1))/2: REM N IS EVEN
120 END
```

6. The mode is used to determine the most popular item. The mode might be used by a buyer of shoes, for example. What is(are) the mode(s) of the following list of numbers?

   3, 2, 7, 6, 3, 7, 2, 5, 1, 2, 2, 3, 7, 3
   (Answers are 2, 3.)

Applications:

1. Write a program to print the median of 20 numbers.

2. Write a program to print the median of 21 numbers.

Summary:

1. What are three types of measures of central tendency?

2. What is the difference between finding the median of an odd number of elements and finding the median of an even number of elements?

3. Under what circumstances would you determine a mode?

Assignments:

1. Write a program to print the arithmetic mean of the marks in your computer class.

2. Input the names and salaries of 25 employees. Print the names and salaries of the employees in ascending order of the salaries.

3. Write the same program as above for 26 employees.

Assignment for More Advanced Students:

1. Write a program to print the mode(s) of a list of numbers.

Topic 63
STATISTICS — MEASURES OF DISPERSION

Performance Objectives: The students will be able to:

- describe the different measures of dispersion
- write a program to determine the range of a set of numbers
- write a program to determine the variance and standard deviation of a set of numbers

Vocabulary: New: measures of dispersion, range, variance, standard deviation

Motivation:

- What is the arithmetic mean of each of the following sets of numbers?

  a) 97, 100, 103
  b) 1, 2, 297

Aim: In what ways can data be analyzed as to its variation from a mean?

Development:

1. As shown by the motivation problem, data which has the same mean may not appear to be similar. The mean does not reflect the fact that numbers may be bunched together. Measures of dispersion provide information as to the variation of the numbers from an arithmetic mean.

2. The measures of dispersion are: a) the range is defined as the difference in value between the highest and lowest value; b) the variance is the sum of the squares of the differences of each number from the arithmetic mean divided by the number of items; c) the standard deviation is the square root of the variance.

3. What is the range of the following list of data?

   12, 50, 4, 110, 22, 37 (Answer is 110-4=106).

   The range is of limited use since it only involves the highest and lowest values but not the values in between. The following program segment might be used to find the range:

```
10 DIM A(50)
20 HIGH=A(1):LOW=A(1)
30 FOR I=2 TO 50
40 IF A(I)>HIGH THEN HIGH=A(I)
50 IF A(I)<LOW THEN LOW=A(I)
60 NEXT I
70 PRINT "RANGE IS ";HIGH - LOW
```

4. The variation of a set of data provides more detail about dispersion than the range since every number in the set of data is involved. The following is a program to print the variance of a list of 40 numbers:

```
10 DIM A(40)
20 REM ARITHMETIC MEAN
30 SUM=0:X=0
40 FOR I=1 TO 40
50 INPUT A(I)
60 SUM=SUM+A(I)
70 NEXT I
80 AVERAGE=SUM/40
90 REM SUM OF DIFFERENCES FROM MEAN SQUARED
100 FOR I=1 TO 40
110 X=X+(AVERAGE-A(I))*(AVERAGE A(I))
120 NEXT I
130 REM VARIANCE
140 PRINT "VARIANCE IS ";X/40
```

5. The standard deviation is used more often than the variance in research reports since the standard deviation is a smaller number.

Application:

1. Write a program to print the standard deviation of a list of 50 numbers.

Summary:

1. How do you find the range of a set of numbers?

2. How do you find the variance of a set of numbers?

3. How do you find the standard deviation of a set of numbers?

Assignments:

1. Write a program to input 50 elements. Print the arithmetic mean, median, range, variance, and standard deviation.

Assignment for More Advanced Students:

1. On standardized tests, scores are often reported as percentile ranks. A percentile rank indicates the percent of students over which that particular rank scored. Write a program to print a percentile table for a list of 300 scores.

Topic 64
SOLVING QUADRATIC EQUATIONS

Performance Objectives: The students will be able to:

- solve quadratic equations using the quadratic formula
- write a program to solve a quadratic equation using the quadratic formula

Vocabulary: New: quadratic equation, roots, quadratic formula, discriminant

Motivation:

- Consider $y = x^2 - x - 12$. For what value(s) of x will y be 0? Answer: x=4,x=-3. The values of x for which y is 0 are called roots of the quadratic equation:

  $0=x^2 - x - 12$

  Solving a quadratic equation means finding the roots.

Aim: How can one solve a quadratic equation?

Development:

1. Consider the quadratic equation in the general form

   $0 = ax^2 + bx + c$. For example, if

   $0= x^2 - x -12$

   then a = 1, b = -1, c = -12. The quadratic formula which is obtained by a method called completing the square can be used to find the roots of a quadratic equation. The quadratic formula is:

   Root 1 = (-B + sqrt(B^2 - 4AC))/2A

   Root 2 = (-B - sqrt(B^2 - 4AC))/2A

   For example, for $0=x^2 - x - 12$:

   Root 1 = (-(-1)+sqrt((-1)-4*1*(-12)))/2

   = 4:

   Root2 = (-(-1)-sqrt((-1)-4*1*(-12)))/2

   = -3.

2. A program segment to determine the roots of a quadratic equation is:

```
10 REM INPUT COEFFICIENTS
20 INPUT A,B,C
30 REM USE QUADRATIC FORMULA TO FIND ROOTS R1,R2
40 R1=(-B+SQRT(B*B-4*A*C))/(2*A)
```

```
50 R2=(-B-SQRT(B*B-4*A*C))/(2*A)
60 PRINT "THE ROOTS OF ";A;"X*X";B;"X + X";C;" ARE ";R1;" AND ";R2
70 GOTO 20
```

3. The value under the square root symbol, B*B-4*A*C, is called the discriminant. If the discriminant is negative, the roots are not real numbers.

Application:

1. Write a program to print a message if the roots of a quadratic equation are not real.

Summary:

1. What is a quadratic equation?

2. What is the quadratic formula?

3. What is the discriminant?

Assignments:

1. Write a program to check the roots obtained by the quadratic formula in the original equation.

2. Write a program to print the roots of quadratic equations to the nearest tenth.

Assignment for More Advanced Students:

1. Write a program to determine the nature of the roots from the discriminant, that is whether the roots are real, rational, irrational, or equal. The program should use the quadratic formula to check the results from the discriminant.

Topic 65
SOLVING POLYNOMIAL EQUATIONS

Performance Objectives: The student will be able to:

- identify polynomial equations
- state the location principle
- write a program to approximate the roots of a polynomial equation

Vocabulary: New: polynomial equations, location principle, equation
Review: quadratic equations, function

Motivation:

- Consider the equation $x^3 - 7x + 6 = 0$. Equations of the form:

$$a_n x^n + \ldots + a_1 x + a_0 = 0$$

  - Equations of this form, where all a's are real numbers and n is a positive integer are called polynomial equations. The degree of a polynomial equation is n. Can ideas derived from solving quadratic equations be used to solve polynomial equations? Polynomial equations often arise in engineering, economics, and science problems.

Aim: How can polynomial equations be solved?

Development:

1. Solve the quadratic equation:

   $x^2 - 3x - 10 = 0$ $(x=5, x=-2)$.

   Look at the function $y = x^2 - 3x - 10$. What are the values of y for $x=4, x=6, x=-3, x=-1$? Notice that the sign of y changes for the x values to the left of root 5 and to the right of root 5. The same is true for the root -2. The y value for x=5 (a root) is 0 and for x=-2 (a root) is 0. Therefore, to locate a root graphically, first find two x values which give y values with opposite signs. The root is located between these two x values. By narrowing the range between these two x values, the root can be approximately located.

2. Note that $y = x^3 - 7x + 6$ is a polynomial function and $0 = x^3 - 7x + 6$ is a polynomial equation. As a quadratic equation has two roots, a cubic equation such as:

   $0 = x^3 - 7x + 6$ has 3 roots. In general, the number of roots of a polynomial equals the degree n.

3. The location principle as applied to graphs of polynomial functions states that for the interval $a \leq x \leq b$, if the y values at x=a and x=b have opposite signs, then there is at least one root of the polynomial equation between x=a and x=b.

4. The algorithm for the program to solve polynomial equations is for consecutive values of x, if the product of the corresponding y values is negative, then at least one root must be between these two x values.

5. A program segment to find one root of the polynomial equation

   $0 = x^3 - 7x + 6$ to the nearest integer is:

```
90 REM TRY VALUES OF X FROM -10 TO +10 IN INCREMENTS OF .1
100 X=-10
110 Y1=X↑3+7*X+6
120 X=X+.1
130 Y2=X↑3+7*X+6
140 IF Y1*Y2<0 THEN 160
150 GOTO 110
160 REM ROUND OFF
170 X=X+.4
180 PRINT "ONE ROOT IS ";INT(X)
```

6. As a check, determine if the corresponding value of y is close to 0 for the value of x.

Application:

1. Write a program to find the remaining two roots of the polynomial equation:

   $0 = x^3 + 7x - 6$

Summary:

1. What is a polynomial equation?

2. What is a polynomial function?

3. What is the location principle?

Assignments:

1. Write a program to solve cubic equations whose coefficients are inputed by the user.

2. Write a program to solve a polynomial equation whose degree and coefficients are inputed by the user.

Topic 66
THE MONTE CARLO METHOD

Performance Objectives: The students will be able to:

- determine the area of a region using the Monte Carlo method
- determine a value for pi using the Monte Carlo method

Vocabulary: New: Monte Carlo method, annulus

Motivation:

- Use the graphics of your computer to draw an irregularly shaped object. Try to determine a method of finding the area of the region.

Aim: How do you use the computer to find the area of objects?

Development:

1. To find the area of a region, the computer can randomly choose points within a larger known area. The number of points within the unknown region can be counted. The following proportion can be used to determine the area of the unknown region:

   ```
   area unknown     # points(unknown)
   ------------ = ------------------
   area known       # points(known)
   ```

   Then, area unknown = (area known * # points(unknown))/# points(known)
   This method is called the Monte Carlo method because the computer is using a game of chance (as is done in Monte Carlo) to solve a problem.

2. The computer can be used to determine if the points fall within the unknown region if there is an equation for the unknown region, as there is for a circle. The algorithm is circumscribe a square about the circle. Use the following proportion:

   ```
   area circle    # of hits
   ----------- = -----------------
   area square    total # of points
   ```

   Then, area circle = (area square * # of hits)/total # of points

3. The program segment is:

   ```
   10 HITS=0
   20 REM USE 100 RANDOM VALUES OF X AND Y (EACH >=0 AND <2)
   30 FOR POINTS=1 TO 100
   40 X=2*RND(1):Y=2*RND(1)
   50 IF (X*X + Y*Y) <=1 THEN HITS=HITS+1
   60 NEXT POINTS
   70 REM FORMULA FOR AREA OF CIRCLE (TOTAL # OF POINTS= 100: AREA SQUARE=4)
   80 AREA=4*(HITS/100)
   90 PRINT "AREA OF CIRCLE IS "; AREA
   ```

Application:

1. Write a program to find the area of an annulus (doughnut shaped figure).

Summary:

1. What is the Monte Carlo method?

2. What is the proportion ln the Monte Carlo method?

Assignments:

1. Write a program to use the Monte Carlo method to find the area of slices (sectors) of a circle, with different central angles.

2. Write a program to use the Monte Carlo method to find the area of pi to 4 decimal places.

Section

9

APPENDICES

Appendix A
COMPUTERS AND SPECIAL EDUCATION STUDENTS

Special educators must prepare students to live in a time when many aspects of community life will be computer-dependent. Daily living skills, such as banking, using the library, and shopping, are rapidly undergoing computerization. Skills taught only a few years ago are, in some cases, already obsolete. Special educators face the challenge of preparing their students to live in a world in which computer literacy may well prove to be as important as book literacy.

Our present special education program attempts to serve students with a wide range of intellectual, motor and communications skills. In recent years there have been tremendous advances in the range and sophistication of adaptive devices to be used with microcomputers. Such devices aid those with special needs in their use of computers.

For the blind, there are Braille keyboards, Braille printers, and programs that convert Braille to English and vice versa. The visually impaired can make use of a device that enlarges the on-screen letters up to sixteen times their normal size.

Software is now available which helps teach lip-reading through animated graphics. Another program depicts the pitch and frequency of the teacher and speech student's voices side by side, and learning can take place using a very accurate feedback method.

A number of control devices have been developed for the physically disabled. Some even allow students with only one physical control site to interface with the microcomputer. The user with good head control can use an optical headpointer; a manual pointer can be utilized by someone with fine motor skills; those with more erratic control can use a wobble switch; for quadriplegics with speech ability, there are voice-activated systems.

By maneuvering one switch in any of the above fashions, the user can perform quite sophisticated functions. It is becoming possible to operate the computer with more subtle body cues. Handicapped students who were formerly excluded from the mainstream of education may no longer face obstructions to learning.

There are a great number of ways in which the microcomputer can be utilized in the special education classroom. The computer can be utilized as a type of "electronic blackboard." The teacher inputs data which could be viewed by the entire class on a large monitor. The teacher might also call individual students to input a line of data while the rest of the class looks on. In this way, the computer is utilized as a tool for class participation.

This type of lesson would best be followed by a reinforcement in which students went to separate terminals. The smaller class size of most special education classes makes this economically feasible.

There are several important reasons why a computer curriculum should be utilized by special education. The computer allows students to make errors

in a non-threatening way. It has infinite patience (if so programmed) and is non-judgmental in its feedback. This form of non-threatening feedback may help decrease the great amount of stress which is so evident in many of our special education classrooms. Some research indicates that many learning-disabled students show a decline in impulsive behavior while working at the computer and a concomitant decline of impulsive behavior in other learning activities.

Many of our special education students have limited success with paper and pen activities. The highly motivating and non-threatening aspects of the computer may allow us to reach a few new students. The individualized nature of computer instruction will allow the teacher to spend a greater amount of time working with students on a one-to-one basis. At the same time, individualized software assures that each student at a terminal is receiving relevant work and immediate feedback. Careful selection of software packages will assure that the needs of both the fast and slow learner are met. The computer will allow all students to work at their own pace.

Many special education students display a greater need for repetition and routine than their regular classroom counterparts. Drill and practice, which are often tedious, can be made more exciting through the use of animation, sound effects and game-playing situations. Thus, the need of many of our students for "overlearning" can be met without the usual dread.

Finally, the problem-solving aspects of computer programming allows special students to move out of the role of passive learner into the role of active problem solver.

The educational potential for the computer in the classroom is great, but there are some problems which educators have to deal with. Most computer assisted instruction is transmitted visually. Special education students with reading problems may have difficulty following the written instructions. Therefore, teachers should be circulating around the room and reinforcing the visual cues with verbal directions.

Another potential problem is that a computer-delivered tutorial system, may, by gaining in ability to individualize instruction, lose in teaching social interpersonal skills. Many special educational students have a need to develop more positive social skills. Will the computer kill the change to develop these skills in the classroom? Again, the teacher can make a big difference here. By adding human reinforcement in the form of praise or an approving touch, we can keep the student from becoming too isolated. Teachers may want to allow two or more students to work together on a particular program. This should not be offered as a rationale for having more students per computer. The wide variety of learning styles and individual needs among our special population calls for one child per computer wherever possible. Since the special education class is usually smaller in size than the mainstream class, this is generally not a problem.

A third potential problem is the possibility of using the computer to instruct in areas that could best be presented in another medium. One should not try to teach skills which are best presented auditorily with a computer which relies so heavily on visual material. Advances in the development of the

talking computer and other technological innovations will allow teachers to utilize computers in teaching to a student's strongest modality.

Yet another area of concern is the possible use of the computer as an "electronic babysitter." There are many arcade-type games available for use with micros. While these games can serve as reinforcers in certain instances, one must be careful not to overuse them.

Finally, it seems that the technological innovations in computer hardware have outdistanced the development of appropriate software in the special education field. There is a great challenge for special education teachers to develop software which will serve not only as electronic textbooks but will maximize their students' potential.

In general, software systems designed for special education should have the following characteristics:

1. Diagnostic and prescriptive strategies which will place the user at the correct level.

2. It should be organized into a hierarchy of skills.

3. It should have clear performance objectives which are measureable.

4. It should be individualized and provide immediate feedback to user responses.

5. It should utilize a multi-sensory approach.

Teachers who have little experience in programming can still be of great help in developing good computer instruction programs. First, it is important for them to preview hardware that is going to be purchased by their schools. Will it meet the needs of the special students in their classrooms? Furthermore, teachers need to review and constructively criticize computer software in much the same way they evaluate other curricular material.

Appendix B
SUGGESTIONS FOR FURTHER READING

This list of books contains only well-written volumes that are suitable as textbooks or classroom reference works. It does not include reference works for languages other than assembler language, BASIC, and Logo, because only these are available free with the current computers. This market changes so quickly that it's wise to check the publishers' current catalogs and, if possible, examine books directly before making any purchases.

BASIC

BASIC
Samuel L. Marateck
Academic Press

BASIC for Beginners
Gary G. Bitter
McGraw-Hill

BASIC Computer Programming
Thomas C. Bartee
Harper & Row

BASIC: A Hands-on Method
Herbert D. Peckham
McGraw-Hill

BASIC: An Introduction to Computer Programming
Robert J. Bent
Brooks/Cole

BASIC and the Personal Computer
Dwyer and Kritchfield
Addison-Wesley

BASIC Programming for Computer Literacy
David Morsund
McGraw-Hill

A Bit of BASIC
Thomas A. Dwyer
Addison-Wesley

Communicating with the Computer: BASIC
Zeney P. Jacobs
Allyn and Bacon

Computer Programming in the BASIC Language
Neal Golden
Harcourt Brace Jovanovich

Computer Programming in the BASIC Language
Zeney P. Jacobs
Allyn and Bacon

57 Practical Programs and Games in BASIC
Ken Tracton
Radio Shack

A Guide to BASIC Programming
Donald D. Spencer
Addison-Wesley

A Guided Tour of Computer Programming in BASIC
Thomas A. Dwyer
Houghton Mifflin

Information Processing with BASIC (4th Ed.)
Marilyn Bohl
Science Research Associates

Introduction to Computer Programming
Crawford and Copp
Houghton Mifflin

More BASIC Computer Games
David Ahl
Creative Computing

More TRS-80 BASIC
Don Inman
John Wiley

The Most Popular Subroutines in BASIC
Ken Tracton
Tab Books

My Computer Likes Me when I Speak in BASIC
Bob Albrecht
Dilithium Press

Problem Solving and BASIC: A Modular Approach
Gustavson and Sackson
Science Research Associates

Programming in BASIC for Business (2nd Ed.)
Bosworth and Nagel
Science Research Associates

Reprinted from "Computers in the Classroom," SRA Publishing Co., Radin and Lee, with permission of the publisher.

Programming with BASIC
Byron S. Gottfried
McGraw-Hill

Programming Techniques for Level II BASIC
William Barden, Jr.
Radio Shack

Thirty-two BASIC Programs for the Apple Computer
Tom Rugg
Dilithium Press

TRS-80 BASIC
Bob Albrecht
John Wiley

TRS-80 Programming in Style
Thomas Dwyer
Radio Shack

TRS-80 Programs
Tom Rugg
Dilithium Press

What Is a Computer?
Marion J. Ball
Houghton Mifflin

LOGO

Apple Logo Primer
Gary G. Bitter and Nancy R. Watson
Reston

Discovering Apple Logo
David D. Thornburg
Addison-Wesley

Introducing Logo for the Apple II, TI99/4A and Tandy Color Computer
Peter Ross
Addison-Wesley

Learning with Apple Logo
Daniel Watt
McGraw-Hill

Learning with Logo
Daniel Watt
McGraw-Hill

Logo Discoveries
Margaret L. Moore
Creative Publications

Logo for the Apple II
Harold Abelson
Byte/McGraw-Hill

Mindstorms
Seymour Papert
Basic Books Inc.

1, 2, 3 My Computer and Me
Donna Bearden
Reston

Turtle Geometry
Harold Abelson and Andrea diSessa
MIT Press

MACHINE LANGUAGE

Assembly Level Programming for Small Computers
Walter J. Weller
D. C. Heath

The 8080a Bugbook
Peter R. Rony
Howard W. Sams

8080/8085 Assembly Language Programming
Lance A. Leventhal
Osborne and Associates

8080/Z80 Assembly Language
Alan R. Miller
John Wiley

How to Program the Z80
Rodnay Zaks
Sybex, Inc.

Microcomputer Primer
Mitchell Waite
Howard W. Sams

Programming Microprocessors
M. W. McMurran
Tab Books

TRS-80 Assembly Language
Hubert S. Howe, Jr.
Prentice-Hall

TRS-80 Assembly Language Programming
William Barden, Jr.
Radio Shack

Understanding Microprocessors
Don L. Cannon
Texas Instruments, Inc.

Z80 Assembly Language Programming
Lance A. Leventhal
Osborne and Associates

Z80 and 8080 Assembly Language Programming
Kathe Spracklen
Hayden

Z80 Instruction Handbook
Nat Wadsworth
Scelbi Publications

Z80 Microcomputer Handbook
William Barden, Jr.
Howard W. Sams

Z80 Programming for Logic Design
Adam Osborne
Osborne and Associates

Z80 Software Gourmet Guide and Cookbook
Nat Wadsworth
Scelbi Publications

GENERAL

An Apple for Kids
Sharon Boren
Dilithium Press

An Apple in the Classroom
Sharon Boren
Dilithium Press

Apple II User's Guide
Poole, McNiff and Cook
McGraw-Hill

The Architecture of Microcomputers
S. E. Greenfield
Winthrop

An Atari in the Classroom
Sharon Boren
Dilithium Press

Automatic Data Processing Systems
Gregory and Van Horn
Wadsworth Publishing Co.

The Compleat Computer (2nd Ed.)
Van Tassel and Van Tassel
Science Research Associates

The Computer Age
Michael L. Dertouzos
MIT Press

Computer Dictionary and Speller
Charles J. Sippl with JoAnne Mayer
Spectrum Books

Computer Literacy for School Administrators and Supervisors
Radin and Greenberg
Lexington Books (D. C. Heath)

Computer Science: A Primer
Forsythe
John Wiley

Computer Systems Performance Evaluation
Israel Borovits
Lexington Books (D. C. Heath)

Computer Usage Fundamentals
Eric A. Weiss
McGraw-Hill

Computers and Reading Instruction
Leo D. Geoffrion and Olga P. Geoffrion
Addison-Wesley

Computers in Business (4th Ed.)
Donald H. Sanders
McGraw-Hill

Computers Today
Donald H. Sanders
McGraw-Hill

Courseware in the Classroom
Ann Lathrop and Bobby Goodson
Addison-Wesley

Data Base Computers
Olin H. Bray and Harvey A. Freeman
Lexington Books (D. C. Heath)

Data Structures and Computer Architecture
Kenneth J. Thurber and Peter C. Patton
Lexington Books (D. C. Heath)

A Dictionary of Computers
Anthony Chandor
Penguin Books

Digital Computer Fundamentals
Thomas C. Bartee
McGraw-Hill

The Elements of Programming Style
Kernighan and Plauger
McGraw-Hill

Flowcharting
McQuigg and Harness
Houghton Mifflin

Goal Programming and Extensions
James P. Ignizzio
Lexington Books (D. C. Heath)

Home Computers: A Beginner's Glossary and Guide
Merl K. Miller and Charles J. Sippl
Dilithium Press

Home Computers: Hardware
Rich Didday
Dilithium Press

Home Computers: Software
Rich Didday
Dilithium Press

How to Profit from Your Personal Computer
T. G. Lewis
Hayden Book Co., Inc.

IBM Personal Computer
Larry Goldstein and Martin Goldstein
Prentice-Hall

The Information Age
Davis and McCormack
Addison-Wesley

Instructional Computing for Today's Teachers
E. Vockell and Robert H. Rivers
Macmillan

Learning and Teaching with Computers
Tim O'Shea and John Self
Prentice-Hall

Microcomputer Resource Book for Special Education
Dolores Hagen
Reston

Microcomputers and the Three R's
Christine Doerr
Hayden Book Co., Inc.

Microcomputers in Education
Edited by Katherine Clay
Oryx Press

Microprocessors and Microcomputer Systems
Dwight H. Sawin III
Lexington Books (D. C. Heath)

Mindstorms
Seymour Papert
Basic Books

Pathways through the ROM
George Blank
Softside Publications

A Pet in the Classroom
Sharon Boren
Dilithium Press

PET/CBM Personal Computer Guide
Osborne and Donahue
Osborne and Associates

Practical Guide to Computers in Education
Peter Coburn
Addison-Wesley

Principles of Business Data Processing (4th Ed.)
Dock and Essick
Science Research Associates

Problems for Computer Solution
Donald D. Spencer
Hayden

Putting Computer Power in Schools
Jerry L. Patterson and Janice H. Patterson
Prentice-Hall

Real-Time Programming with Microcomputers
Ronald C. Turner
Lexington Books (D. C. Heath)

Social Effects of Computer Use and Misuse
J. Mack Adams
John Wiley

Social Issues in Computing
C. C. Gottlieb
Academic Press

Software Design for Microcomputers
Carol Anne Ogdin
Prentice-Hall

Using the TRS-80 in Your Home
Charles D. Sternberg
Radio Shack

Appendix C
GLOSSARY

Array An array is a collection of items that are all labeled with the same *variable* but have different *subscripts*. A(1), A(2), A(3), A(4), and A(5) make up an array of variables; they are all named A, but A(1) is the first A, A(2) is the second A, and so on. An expression such as A(E), where E is being used to count errors (for example), is also an array, because the value of E will increase with each *iteration*. So A(E) will be, successively, A(0), A(1), A(2), etc.

Binary Binary numbers are based on the number 2, instead of the usual base of 10. Binary place values are 1, 2, 4, 8, 16, 32, etc., instead of the more familiar 1, 10, 100, 1000, etc. With this system all numbers can be written with two symbols, 0 and 1, instead of the usual ten sumbols (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9). All typewriter characters can be translated into binary, via the ASCII code. Computers perform all manipulations, both of words and of numbers, in binary arithmetic.

Bit A bit is a single digit in the *binary* code. Both numbers and letters are encoded as BInary digiTs—hence, bits.

Bug A bug is a defect. This word generally refers to an incorrect statement or sequence of statements in a computer *program*. (Chapter 1)

Byte The entire *binary* representation of a character is called a byte. In the computer's binary code the letter *A* looks like 1000001, *B* looks like 1000010, and so on. The whole expression is a byte; each individual digit is a *bit*.

CAI Computer-assisted instruction uses the computer for drill or practice of specific, factual information. Essentially, CAI employs a computer as an electronic workbook.

Compiler A compiler is a *program* that converts a more human-like computer language, such as BASIC or COBOL, into the computer's own *machine language*. A compiler converts an entire program into machine code all at once and the user can then either save or run the resulting machine code.

CRT (Cathode-ray tube) This is a standard television picture tube, the most common display device used with microcomputers.

Cursor The cursor is a visual character that a computer prints on its screen to show where the next piece of information will appear. As you type, your letters are displayed on the screen, and the cursor moves along to show you where your next input will be displayed.

Data Data is information that a computer holds or manages: for example, a mailing list or a *program*. Any information the user asks a machine to remember is called data.

File A file is a stored collection of *data*. Computers can store information either internally in memory or on external devices.

Graphics Graphics is the visual material displayed on the *CRT* screen or output on paper. Cartoon characters, pencil drawings, and even ordinary typewriter characters are forms of computer graphics.

Integrated circuit chip The invention of the integrated circuit chip set off the whole microcomputer revolution. The chips themselves are small plastic, metal, and silicon devices that contain thousands of separate storage locations. They have replaced the tubes and transistors used in earlier machines, since they are both faster and more durable. A single silicon chip about one inch long can store many thousands of pieces of information. There are chips now in production that can each hold more than a million facts.

Reprinted from "Computers in the Classroom," SRA Publishing Co., Radin and Lee, with permission of the publisher.

Interface An interface is a piece of hardware used to connect two devices, to translate between them. Each manages *data* in a different way. A printer, for example, moves data very slowly but a CPU moves data much more quickly. An interface allows the two to exchange data by taking the data in at one speed and sending it out at another.

Interpreter Like a *compiler*, an interpreter is a *program* that translates the more human-like computer languages into the machine's code. Interpreters are slower than compilers, since they translate each instruction every time it is used. Interpreters usually require much less memory space than compilers do.

I/O devices Input/output devices are pieces of hardware that route information into and out of the CPU. Disk drives can serve both input and output functions, but the keyboard and *CRT* screen are unidirectional I/O devices.

Iteration Iteration involves counting. Each time the computer repeats the series of steps that form a given operation constitutes one iteration.

K K stands for "thousands." When a computer company says that their machine has 48K of memory capacity, they are telling you that the machine can hold approximately 48,000 characters (or *bytes*) in memory at one time. (Actually 1K = 1024 *not* 1000, but 1000 is close enough for our purposes.)

Machine language Machine language is the code that computers actually use to perform their operations. It consists of commands in the form of binary numbers that cause the computer's electronic switches to open and close.

Modem The MOdulator/DEModulator converts information from computer-readable *binary* digits into sound and then back again. It is used to interface a computer to a telephone line, so that one computer can talk to another a great distance away.

Monitor program The monitor program comes in the *ROM* of the computer as purchased. It directs the machine's response to keyboard strokes and its transmission of *data* to output devices. It thus provides the CPU with orderly connections to its input and output devices.

Numeric Numeric means "relating to numbers." A numeric *variable* is a label that is affixed to a number when it is stored in computer memory.

Peripherals Parts of a computer that work in connection with it but are not part of it. Devices like tape recorders, disk drives, printers, microphones, and speakers are often used as peripheral equipment.

Program A collection of *statements* that tells a computer how to do a specific job.

Program documentation Program documentation is material that clarifies the nature of a *program*. It comes in two forms: External documentation is the printed matter—manuals, diagrams, and so on—that describes the parameters of a program and, essentially, guides the user through the required interactions. Internal documentation is actually part of the program. Within this category there are two subgroups: user documentation and programmer documentation. The first consists of the PRINT *statements* that tell the user what to expect as the program flows through its course. The second comprises the REMark statements used by the programmer to label different sections of the program.

RAM Random access memory is the space where the computer stores *programs* and other *data*. Material in RAM can be erased and is accessible to programmers.

ROM Read-only memory is information that is stored in a computer and cannot easily be altered. *Compilers* and *interpreters* are often stored in ROM, which is also called "hardwired software" or "firmware."

Simulation A *program* that allows a user to test solutions to "real life" problems. The user suggests an answer to the problem and the computer projects the consequences. Simulation allows users to test out different approaches to a situation on a machine, without the risks and difficulties of experimenting in the real world.

Software Software refers to sets of instructions with which a programmer causes the computer to perform specific tasks.

Statement A statement is a computer instruction of one or more words that tells the computer to do a specific task. A sequence of statements forms a *program*.

String A string is a series of keyboard characters—a word or a collection of words and numbers—that makes up a piece of information. Street addresses or even whole sentences or paragraphs can be strings.

String variable A string variable is the label that is affixed to a *string* when it is stored in memory. The computer programmer can access such stored information by using the appropriate variable name.

Subroutine A subroutine is a portion of a program that a programmer needs to access repeatedly. Instead of writing these instructions over each time they are needed, the programmer can put them in a special place in the program and ask the computer to go there, implement the subroutine *statements*, and then return to where it left off (its "calling spot") in the *program*. This technique greatly reduces the size of many programs.

Subscripted variable This is another name for an *array* variable.

User A user is a person who uses a computer. Computer users, including those in the field of education, often know very little about computers, but they do know how to follow directions. They simply turn on their machines and follow the English instructions provided by a prepackaged *program*.

Variable A variable is a label affixed to words or numbers that you wish to store in the computer. The computer remembers not only the *data* but also the variable that identifies them, so that you can retrieve or change the information later.

Word processing Writing documents with the aid of a computer. We wrote this book with a computer and a word-processing *program*. It enabled us to type in the text, modify it, store it, and print it out in a formatted (i.e., standardized) fashion. Word processing reduces the time spent typing repetitive documents; it can cut in half the time spent on secretarial functions.

Appendix D
EDUCATIONAL SOFTWARE VENDORS

The vendors listed here all supply high-quality educational software. We couldn't test every piece of software from every vendor, but we have examined enough of each vendor's material to convince ourselves of the overall quality. Because of the evolving nature of this field, we recommend that you test software yourself before purchasing it. Programs that work well in one set of conditions often won't work at all in another.

Academic Computing Association
P.O. Box 27561
Phoenix, Ariz. 85061

Academic Software
c/o Software City
22 East Quackenbush Avenue
Dumont, N.J. 07628

Acorn Software Products, Inc.
634 North Caroline Avenue, S.E.
Washington, D.C. 20003

Activity Resources Co., Inc.
P.O. Box 4875
Hayward, Calif. 94540

Addison-Wesley Publishing Co.
2725 Sand Hill Road
Menlo Park, Calif. 94025

Advanced Computer Products, Inc.
P.O. Box 17329
Irvine, Calif. 92713

Adventure International, Inc.
P.O. Box 729
Casselberry, Fla. 32707

Alternate Source
704 Pennsylvania Street
Lansing, Mich. 48906

American Micro Media
P.O. Box 306
Red Hook, N.Y. 12571

Analysts International Corporation
295 Madison Avenue, Suite 930
New York, N.Y. 10017

Aquarius Publishers, Inc.
P.O. Box 128
Indian Rocks Beach, Fla. 33535

Aries Information Systems, Inc.
7400 Metro Boulevard, Suite 370
Minneapolis, Minn. 55435

Avant-Garde Creations
P.O. Box 30160
Eugene, Ore. 97403

Avas Corporation
196 Holt Street
Hackensack, N.J. 97602

Basics and Beyond, Inc.
P.O. Box 10
Amawalk, N.Y. 10501

Betamax, Inc.
101 Nickerson, Suite 550
Seattle, Wash. 98109

Bluebird's Computer Software
2267 23rd Street
Wyandotte, Mich. 48192

Borg-Warner Educational Systems
600 West University Drive
Arlington, Ill. 60004

Brain Box
601 West 26th Street
New York, N.Y. 10001

Camelot Publishing Company
P.O. Box 1357
Ormond Beach, Fla. 32074

Carolina Biological Supply Company
Burlington, N.C. 27215

The Center for Humanities
Communications Park, Box 1000
Mount Kisco, N.Y. 10549

Chang Laboratories
10228 North Stelling Road
Cupertino, Calif. 95014

Charles Clark Co., Inc.
168 Express Drive South
Brentwood, N.Y. 11717

Charles Mann and Associates
55722 Santa Fe Trail
Yucca Valley, Calif. 92284

Reprinted from "Computers in the Classroom," SRA Publishing Co., Radin and Lee, with permission of the publisher.

The Classroom Answer
Houghton Mifflin
One Beacon Street
Boston, Mass. 02107

Classroom Consortia Media
28 Bay Street
Staten Island, N.Y. 10301

The Code Works
Box 550
Goleta, Calif. 93116

Cognitive Research Group
Education Development Center, Inc.
55 Chapel Street
Newton, Mass. 02160

COIN
1546 Dartford Road
Maumee, Ohio 43537

Comm Data Systems, Inc.
P.O. Box 325
Milford, Mich. 48042

Compak, Inc.
10309 Talleyrand Drive
Austin, Tex. 78750

Compress
Division of Science Books International
P.O. Box 102
Wentworth, N.H. 03282

Compu-Tations, Inc.
P.O. Box 502
Troy, Mich. 48099

Computer Cat
3005 West 74th Avenue
Westminster, Colo. 80030

Computer Concepts
509 Chestnut Street
Cedarhurst, N.Y. 11516

Computer Courseware Services
Changing Times Education Service
300 York
St. Paul, Minn. 55101

Computer Discount of America, Inc.
West Milford Mall
West Milford, N.J. 07480

Computer Motivated Learning Laboratory
Random House School Division
201 East 50th Street
New York, N.Y. 10022

Computer Systems International, Inc.
539 Durie Avenue
Closter, N.J. 07624

Conduit
University of Iowa
100 Lindquist Center
P.O. Box 338
Iowa City, Iowa 52244

Courseware Magazine
4919 West Millbrook #222
Fresno, Calif. 93726

Creative Computing
39 East Hanover Avenue
Morris Plains, N.J. 07950

Creative Publications
3977 East Bayshore Road
P.O. Box 10328
Palo Alto, Calif. 94303

Cross Educational Software
P.O. Box 1536
Ruston, La. 71270

Cuisenaire Company of America, Inc.
12 Church Street, Box D
New Rochelle, N.Y. 10805

Curriculum Applications
P.O. Box 264
Arlington, Mass. 02174

Data Command
P.O. Box 548
Kankakee, Ill. 60901

Developmental Learning Materials
One Elm Park
P.O. Box 4000
Allen, Tex. 75002

Dilithium Software
P.O. Box 606
Beaverton, Ore. 97075

Disney Electronics
6153 Fairmount Avenue
San Diego, Calif. 92120

Dolphin Computer-Based Instruction
TSC, a Houghton Mifflin Company
Box 683
Hanover, N.H. 03755

Dorsett Educational Systems, Inc.
Box 1226
Norman, Okla. 73070

Dynacomp, Inc.
1427 Monroe Avenue
Rochester, NY. 14618

George Earl
1302 South General McMullen
San Antonio, Tex. 78237

Earthware Computer Services
P.O. Box 30039
Eugene, Ore. 97403

Edco
P.O. Box 30846
Orlando, Fla. 32862

Educational Activities, Inc.
P.O. Box 392
Freeport, N.Y. 11520

Educational Audio-Visual, Inc.
Pleasantville, N.Y. 10570

Educational Computing Systems, Inc.
106 Fairbanks
Oak Ridge, Tenn. 37830

Educational Connection
1508 Coffee Road, Suite J
Modesto, Calif. 95355

Educational Courseware
3 Nappa Lane
Westport, Conn. 06880

Educational Development Corporation
P.O. Box 45663
Tulsa, Okla. 74145

Educational Dimensions Group
P.O. Box 126
Stamford, Conn. 06904

Educational Materials & Equipment Company
P.O. Box 17
Pelham, N.Y. 10803

Educational Micro Systems, Inc.
P.O. Box 471
Chester, N.J. 07930

Educational Services Management Corporation
Software Support Division
P.O. Box 12599
Research Triangle Park, N.C. 27709

Educational Software, Inc.
Soquel, Calif. 95073

Educational Software Midwest
414 Rosemere Lane
Maquoketa, Iowa 52060

Educational Systems Software
23720 El Toro Road, Suite C
P.O. Box E
El Toro, Calif. 92630

Educational Teaching Aids
159 West Kinzie Street
Chicago, Ill. 60610

Edu-Soft
4639 Spruce Street
Philadelphia, Pa. 19139

Edutech, Inc.
634 Commonwealth Avenue
Newton Centre, Mass. 02159

Eduware Services, Inc.
28035 Dorothy Drive
Agoura, Calif. 91301

Electronic Courseware Systems, Inc.
Box 2374, Station A
Champaign, Ill. 61820

Encyclopedia Britannica Educational Company
425 North Michigan Avenue
Chicago, Ill. 60611

Entelek
P.O. Box 1303
Portsmouth, N.H. 03801

Evans Newton, Inc.
7745 East Redfield Road
Scottsdale, Ariz. 85260

Eye Gate Media, Inc.
146-01 Archer Avenue
Jamaica, N.Y. 11435

Fisher Scientific Company
Educational Materials Division
4901 West Lemoyne Street
Chicago, Ill. 60651

Follett Library Book Company
4506 Northwest Highway
Crystal Lake, Ill. 60014

Foreign Language CAI Software
Gessler Publishing Company
900 Broadway, Suite 10A
New York, N.Y. 10003

Fullmer Associates
1132 Via Jose
San Jose, Calif. 95120

Gamco Industries, Inc.
Box 310-P
Big Spring, Tex. 79720

George Madden & Associates
3101 Fourth Avenue
San Diego, Calif. 92103

Gladstone Electronics
901 Fuhrmann Boulevard
Buffalo, N.Y. 14203

J. L. Hammett Company
Box 545
Raintree, Mass. 02184

Harcourt Brace Jovanovich
School Department
757 Third Avenue
New York, N.Y. 10017

Harper & Row/Conduit
110 East 53rd Street
College Division, Suite 3D
New York, N.Y. 10022

Hartley Courseware, Inc.
Dept. CND, Box 431
Dimondale, Mich. 48821

Hayden Educational Software
50 Essex Street
Rochelle Park, N.J. 07662

H & E Computronics, Inc.
50 North Pascack Road
Spring Valley, N.Y. 10977

Holt, Rinehart, and Winston
383 Madison Avenue
New York, N.Y. 10017

Houghton Mifflin Microcourse
Language Arts
One Beacon Street
Boston, Mass. 02107

Houghton Mifflin Microcourse
Reading
One Beacon Street
Boston, Mass. 02107

ICT-Instructional/Communications
Technology, Inc.
10 Stepar Place
Huntington Station, N.Y. 11746

Ideatech Company
P.O. Box 62451, Dept. 88
Sunnyvale, Calif. 94088

Information Unlimited Software, Inc.
281 Arlington Avenue
Berkeley, Calif. 94707

Instant Software
Peterborough, N.H. 03458

Instructional Development Systems
1927 Virginia Beach Boulevard
Virginia Beach, Va. 23452

Interpretive Education, Inc.
157 South Kalamazoo Mall
P.O. Box 3126
Kalamazoo, Mich. 49003

I.S.A., Inc.
P.O. Box 7186
Wilmington, Del. 19803

Jadee Enterprises
1799 Meadowlake
Charleston, Ill. 61920

Jinsam Micro-Systems, Inc.
Box 274E
Kingsbridge Station
Riverdale, N.Y. 10463

J & S Software
140 Reid Avenue
Port Washington, N.Y. 11050

K8 Software
P.O. Box 248
Canton, Conn. 06019

Kensington Microware
300 East 54th Street
New York, N.Y. 10022

Krell Software
1320 Stony Brook Road, Suite 219
Stony Brook, N.Y. 11790

K-12 Micromedia, Inc.
P.O. Box 17
Valley Cottage, N.Y. 10989

Lanier Learning Center
1700 Chantilly Drive N.E.
Atlanta, Ga. 30324

The Learning Company
4370 Alpine Road
Portola Valley, Calif. 94025

Learning Systems, Ltd.
P.O. Box 9046
Fort Collins, Colo. 80525

Learning Tools, Inc.
686 Massachusetts Avenue
Cambridge, Mass. 02139

Level IV Products, Inc.
32461 Schoolcraft Street
Livonia, Mich. 48150

Magic Lanterns Computers
406 South Park Street
Madison, Wisc. 53715

Marck
280 Linden Avenue
Branford, Conn. 06405

Math Software
1233 Blackhorn Place
Deerfield, Ill. 60015

Mathware/Math City
4040 Palos Verdes Drive North
Rolling Hills Estates, Calif. 90274

MCE Educational Programs
Interpretive Education, Inc.
157 South Kalamazoo Mall
Kalamazoo, Mich. 49007

McGraw-Hill Technology Marketing Manager
School Division
McGraw-Hill Book Company
1221 Avenue of the Americas
New York, N.Y. 10020

MEAN
Microcomputer Education Applications Network
256 North Washington Street
Falls Church, Va. 22046

MECC
Minnesota Educational Computing Consortium
2520 Broadway Drive
St. Paul, Minn. 55113

Media Materials, Inc.
2936 Remington Avenue
Baltimore, Md. 21211

Med Systems Software
P.O. Box 2674
Chapel Hill, N.C. 27514

Meka Publishing Company
9120 Galaxie Drive
Indianapolis, Ind. 46227

Mercer Systems, Inc.
87 Scooter Lane
Hicksville, N.Y. 11801

The Micro Center
P.O. Box 6
Pleasantville, N.Y. 10570

Microcomputers Corporation
P.O. Box 8
Armonk, N.Y. 10504

Microcomputer Workshops
103 Puritan Drive
Port Chester, N.Y. 10573

Micro-Ed., Inc.
P.O. Box 24156
Minneapolis, Minn. 55424

Micrograms, Inc.
P.O. Box 2146
Loves Park, Ill. 61130

Micro Lab, Inc.
2310 Skokie Valley Road
Highland Park, Ill. 60035

Micro Learningware
P.O. Box 2134
North Mankato, Minn. 56001

Microphys Programs
2048 Ford Street
Brooklyn, N.Y. 11229

Scholastic Microcomputer Materials
Scholastic, Inc.
904 Sylvan Avenue
Englewood Cliffs, N.J. 07632

Science Research Associates, Inc.
155 North Wacker Drive
Chicago, Ill. 60606

Scott, Foresman & Company
1900 East Lake Avenue
Glenview, Ill. 60025

SEI (Slina Enterprises, Inc.)
P.O. Box 7266-CD
Hampton, Va. 23666

Sled Software
P.O. Box 16322
Minneapolis, Minn. 55416

The Software Exchange
6 South Street
Milford, N.H. 03055

The Software House, Inc.
695 East Tenth Street North
Logan, Utah 84321

Spinnaker Software Corporation
215 First Street
Cambridge, Mass. 02142

Sterling Swift Publishing Company
1600 Fortview Road
Austin, Tex. 78704

Storybooks of the Future
527-41st Avenue
San Francisco, Calif. 94121

Sunburst Communications
39 Washington Avenue, Room VF #4
Pleasantville, N.Y. 10570

S.V.E., Inc.
1345 Diversey Parkway
Chicago, Ill. 60614

Synergistic Software
5221 120th Avenue, S.E.
Bellevue, Wash. 98006

Tara, Ltd.
P.O. Box 118
Seiden, N.Y. 11784

Teacher's Pet
1517 Holly Street
Berkeley, Calif. 94703

Teaching Tools: Microcomputer Services
P.O. Box 50065
Palo Alto, Calif. 94303

Teach Yourself by Computer Software (TYC)
40 Stuyvesant Manor
Genesco, N.Y. 14454

Teck Associates
P.O. Box 8732
White Bear Lake, Minn. 55110

Telephone Software Connection, Inc.
P.O. Box 6548
Torrance, Calif. 90504

Terrapin, Inc.
380 Green Street
Cambridge, Mass. 02139

Texas Instruments
P.O. Box 10508
Mail Station 5849
Lubbock, Tex. 79408

T.H.E.S.I.S.
P.O. Box 147
Garden City, Mich. 48135

3R Software
P.O. Box 3115
Jamaica, N.Y. 11431

T.I.E.S.
1925 West County Road, BZ
St. Paul, Minn. 55113
Attn: Distribution

Triangle Audio-Visual Media, Inc.
141 North Main Street
P.O. Box 2248
Branford, Conn. 06405

TSE Hardside
6 South Street
Milford, N.H. 03055

Tycom Associates
68 Velma Avenue
Pittsfield, Mass. 01201

Micro Power and Light Company
12820 Hillcrest Road #224
Dallas, Tex. 75230

Microsoftware Services
P.O. Box 776
Harrisonburg, Va. 22801

Midwest Visual Equipment Company
6500 North Hamlin Avenue
Chicago, Ill. 60645

Milliken Publishing Company
1100 Research Boulevard
St. Louis, Mo. 63132

Milton Bradley Educational Division
443 Shaker Road
East Longmeadow, Mass. 01028

Monument Computer Service
Village Data Center
P.O. Box 603
Joshua Tree, Calif. 92252

MSSSD, Inc.
3412 Binkley
Dallas, Tex. 75205

Muse Software
330 North Charles Street
Baltimore, Md. 21201

Opportunities for Learning, Inc.
8950 Lurline Avenue
Dept. A7401
Chatsworth, Calif. 91311

Orange Cherry Media
7 Delano Drive
Bedford Hills, N.Y 10507

The Oryx Press
2214 North Central
Phoenix, Ariz. 85004

Powersoft
P.O. Box 157
Pitman, N.J. 08071

Prescription Learning
5240 South Sixth Street
Springfield, Ill. 62707

Prismatron Productions, Inc.
155 Buena Vista Avenue
Mill Valley, Calif. 94941

Program Design, Inc.
11 Idar Court
Greenwich, Conn. 06830

The Programmers, Inc.
P.O. Box 1207
211 Cruz Alta
Taos, N.M. 87571

Programs Unlimited
P.O. Box 265
Jerico, N.Y. 11753

Programs Unlimited
131 Mamaroneck Avenue
White Plains, N.Y. 10605

Quality Educational Designs
P.O. Box 12486
Portland, Ore. 97212

Queue, Inc.
5 Chapel Hill Drive
Fairfield, Conn. 06432

Radio Shack Education Division
Dept. 82-A-324
1300 One Tandy Center
Ft. Worth, Tex. 76102

Random House School Division
201 East 50th Street
New York, N.Y. 10022

Random House School Division
400 Hahn Road
Westminster, Md. 21157

Reader's Digest Services, Inc.
Educational Division
Pleasantville, N.Y. 10570

Reston Publishing Company
11480 Sunset Hills Road
Reston, Va. 22090

Right-on Programs
Division of Computeam, Inc.
P.O. Box 977
Huntington, N.Y. 11743

RMI Media Productions, Inc.
120 West 72nd Street
Kansas City, Mo. 64114

Scandura Training Systems, Inc.
1249 Greentree Lane
Narberth, Pa. 19072

TYC Software
40 Stuyvesant Manor
Genesco, N.Y. 14454

Unicom
297 Elmwood Avenue
Providence, R.I. 02907

Versa Computing, Inc.
3541 Old Condjo Road, Suite 104
Newbury Park, Calif. 91320

Visual Materials, Inc.
4170 Grove Avenue
Gurnee, Ill. 60031

John Wiley and Sons, Inc.
Attn: Clayton Gordon
605 Third Avenue
New York, N.Y. 10158

Wise Owl Workshop
1168 Avenida de las Palmas
Livermore, Calif. 94550

Xtrasoft
P.O. Box 91063
Louisville, Ky. 40291

Zweig Associates
1711 McGraw Avenue
Irvine, Calif. 92714

Appendix E

COMPUTING MAGAZINES

GENERAL EDUCATIONAL COMPUTING MAGAZINES

Classroom Computing News
intentional Educations, Inc.
341 Mount Aubern Street
Watertown, Mass. 02172

Collegiate Microcomputer
Rose-Hulman Institute of Technology
Terre Haute, Ind. 47803

The Computing Teacher
Department of Computer Education
and Information Science
University of Oregon
Eugene, Ore. 97403

Educational Computer Magazine
P.O. Box 535
Cupertino, Calif. 95015

Educational Computing
Oakfield House
Perrymount Road
Haywards Heath, Sussex RH16
3DH England

Educational Technology
140 Sylvan Avenue
Englewood Cliffs, N.J. 07632

Electronic Education
Electronic Communications, Inc.
1311 Executive Center Drive, Suite 220
Tallahassee, Fla. 32301

Electronic Learning
Scholastic, Inc.
902 Sylvan Avenue
Box 2001
Englewood Cliffs, N.J. 07632

Instructional Innovator
AECT
1126 Sixteenth Street, N.W.
Washington, D.C. 20036

Interface: The Computer Education Quarterly
915 River Street
Santa Cruz, Calif. 95060

Journal of Computer-Based Instruction
ADCIS
409 Miller Hall
Western Washington University
Bellingham, Wash. 98225

Pipeline
P.O. Box 388
Iowa City, Iowa 52244

School Microcomputer Bulletin
Learning Publications, Inc.
Box 1326
Holmes Beach, Fla. 33509

T.H.E. Magazine
Information Synergy, Inc.
P.O. Box 992
Acton, Mass. 01720

GENERAL COMPUTER INFORMATION PUBLICATIONS

Byte Magazine
70 Main Street
Peterborough, N.H. 03458

Personal Computing
Hayden Publishing Company, Inc.
50 Essex Street
Rochelle Park., N.J. 07662

MAGAZINES FOR SPECIFIC BRANDS OF COMPUTERS

Compute
Small Systems Services, Inc.
P.O. Box 5406
Greensboro, N.C. 27403
(PET, Atari)

Creative Computing Magazine
P.O. Box 789M
Morristown, N.J. 07960
(TRS-80, Apple, PET, Atari, Sol)

Reprinted from "Computers in the Classroom," SRA Publishing Co., Radin and Lee, with permission of the publisher.

Dr. Dobbs' Journal
People's Computing Company
Box E
1263 El Camino Real
Menlo Park, Calif. 94025
(small computer system users)

80 Microcomputing
1001001, Inc.
80 Pine Street
Peterborough, N.H. 03458
(TRS-80)

Personal Computer
Software Communications, Inc.
1239 21st Avenue
San Francisco, Calif. 94122
(IBM Personal Computer)

Recreational Computing
People's Computing Company
Box E
1263 El Camino Real
Menlo Park, Calif. 94025
(small computer system users)

GENERAL INDUSTRY AND BUSINESS INFORMATION PUBLICATIONS

Mini Microsystems
Cahners Publishing Company
Division of Reed Holdings, Inc.
221 Columbus Avenue
Boston, Mass. 02116

Appendix F
POPULAR HARDWARE VENDORS

The computer field changes so quickly that any list of hardware manufacturers won't stay current for more than a few months. Therefore, we have listed only vendors that have made an impact on the school marketplace and are likely to be around to support their equipment in the foreseeable future.

MICROCOMPUTER MANUFACTURERS

Acorn Computer Corp.
400 Unicorn Pk. Drive
Woburn, Mass. 01801

Apple Computer, Inc.
10260 Bandley Drive
Cupertino, Calif. 95014

Commodore, Inc.
487 Devon Park Drive
Wayne, Pa. 19087

Digital Equipment Corporation
129 Parker Street
Maynard, Mass. 01754

IBM Corporation
Old Orchard Road
Armonk, N.Y. 10504

Monroe Systems for Business
The American Road
Morris Plains, N.J. 07950

Radio Shack
1300 One Tandy Center
Fort Worth, Tex. 76102

Texas Instruments, Inc.
P.O. Box 2909
Austin, Tex. 78769

INTEGRATED-CIRCUIT CHIP MANUFACTURERS

Intel Corporation
3065 Bowers Avenue
Santa Clara, Calif. 95051

Motorola, Inc.
3501 Ed Bluestein Boulevard
Austin, Tex. 78721

Zilog, Inc.
1315 Dell Avenue
Campbell, Calif. 95008

Reprinted from "Computers in the Classroom," SRA Publishing Co., Radin and Lee, with permission of the publisher.

Appendix G

PRIMARY SOURCES FOR GRANT INFORMATION

PRIVATE FOUNDATION GRANTS

There are more than 22,000 private foundations in the United States; any of them can be a source of school grant funds. Each foundation channels its resources toward a limited number of specific programs. Some are only interested in assisting projects that help the handicapped while others involve themselves only in projects that use mass media for education. Your local university and public libraries have many books that can guide you to the proper agency for your grant proposal. *The Foundations Directory* and the *National Data Book*, both published by the Foundation Center in New York City, are very useful guides. You can obtain these books by writing to:

The Foundations Center
888 Seventh Avenue
New York, New York 10106

STATE GRANTS TO EDUCATIONAL INSTITUTIONS

Your State Department of Education is the best source for informational brochures and packets describing the various state grants currently available and the latest deadlines for filing your proposal application. After examining this literature, you can aim your application in the most promising direction. Some state funds are only available to entire school districts (although others are specifically targeted toward individual classrooms), so it is important to read all the available literature carefully before writing your state proposal.

FEDERAL GRANTS TO EDUCATION

The federal government offers a great variety of grants to schools. These grants are available from many different agencies. The daily *Federal Register* can provide you with a potpourri of data about current funding sources and regulations issued by the Department of Education. The *Federal Register* is available in most public and university libraries. The federal government has funded projects as diverse as general remediation, supplementary instruction for handicapped children, enrichment for the gifted and talented, and occupational education furthering integration—to name just a few of the many areas.

We strongly recommend that you write a letter requesting information on official school stationery and send a copy to every federal agency in existence. Agencies frequently change their names, so it is best to first request from the U.S. Department of Education in Washington, D.C., a list of agencies that offer grants to schools. Then you can direct your request for submission kits to the proper source of funds. The address is:

U.S. Department of Education
Washington, D.C. 20202

The best sources of funding are usually local parent and business groups. Don't pass them up!

Reprinted from "Computers in the Classroom," SRA Publishing Co., Radin and Lee, with permission of the publisher.

It is the policy of the New York City Board of Education not to discriminate on the basis of race, creed, national origin, age, handicapping condition, or sex in its educational programs, activities, and employment policies, as required by law. Inquiries regarding compliance with appropriate laws may be directed to Mercedes Nesfield, Director, Office of Equal Opportunity, 110 Livingston Street, Brooklyn, New York 11201; or to Charles Tejada, Director, Office of Civil Rights, Department of Education, 26 Federal Plaza, Room 33-130, New York, New York 10278.